IT/ASD/97-1

LaTeX at CERN

TeX at CERN

Using LaTeX at CERN

(February 1997)

Michel Goossens/IT-ASD

CERN Geneva, Switzerland

Preliminary remarks

This manual presents an introduction to LaTeX 2_ε as it is installed on the reference platforms at CERN, describing CERN-specific extensions, and introducing a few useful packages.

This document has been produced using LaTeX [1] with the `cernman` class, developed at CERN. A gzip compressed PostScript file `texacern.ps.gz`, containing a complete printable version of this manual, can be obtained from any CERN machine by anonymous ftp as follows (commands to be typed by the user are underlined):

```
ftp asisftp.cern.ch
Connected to asis00.cern.ch.
220 asis00 FTP server (Version wu-2.4(2)...) ready.
Name (asisftp:username): ftp
Password: your_mailaddress
230 Guest login ok, access restrictions apply.
ftp> cd cernlib/doc/ps.dir
ftp> get texacern.ps.gz    (type get texacern.ps for the uncompressed version)
ftp> quit
```

Acknowledgements

Sebastian Rahtz (now at Elsevier Science, Oxford, UK) contributed substantially in first setting up the LaTeX system on Unix while working as a consultant at CERN in 1993-94. The present setup relies on Thomas Esser's `teTeX` distribution and uses the TDS directory structure. The whole Unix TeX tree is maintained with the help of the `asis` system developed by Philippe Defert's *et al.* (IT-DCI). Michèle Jouhet, David Stungo and their colleagues of the DTP centres (AS-DH group) look after PC and Mac support. The layout of the VMS system is still mainly based on the one that Alexander Samarin (now at ISO, Geneva) introduced while working at CERN in 1992.

Contact Person Michel Goossens /CN (`goossens@cern.ch`)

Edition – February 1997

Table of Contents

List of Figures

List of Tables

Chapter 1: General introduction to LaTeX at CERN

TeX [11, 12] (pronounced `Tekh`) is a text processing system developed by Donald E. Knuth of Stanford University to compose by computer high quality documents, especially those containing many mathematical formulae. Its typographic quality is comparable to the finest works of the printing art. The TeX text formatting system offers several tens of custom-made fonts and over 300 basic and 600 "composed" `plain` commands.

Unlike most desktop publishing systems, TeX does not attempt to show the appearance of the finished document on the screen as it is edited (although screen previewers are available). Instead, the document is "marked up" with codes, which indicate boldface, italics, special characters (e.g., "`\int`" for an integral sign), and the like. Large-scale aspects of design are automated; you just give the title of a chapter, and let the computer take care of numbering the chapter and putting the title in the right place on the page.

TeX is generally considered the most sophisticated computer typesetting system, as well as (for an experienced user) one of the easiest to use. LaTeX is a macro package build on top of TeX.

Leslie Lamport, the original author of LaTeX, says the following about his system in the preface of [1]

> "LaTeX adds to TeX a collection of commands that simplify typesetting by letting the user concentrate on the structure of the text rather than on formatting commands. In turning TeX into LaTeX, I have tried to convert a highly-tuned racing car into a comfortable family sedan. The family sedan isn't meant to go as fast as a racing car or be as exciting to drive, but it's comfortable and gets you to the grocery store with no fuss. However, the LaTeX sedan has all the power of TeX hidden under its hood, and the more adventurous driver can do everything with it that he can with TeX."

Fundamental to LaTeX is the idea of a *document class* which determines exactly how a document will be formatted. LaTeX provides standard document styles that describe how standard logical elements should be printed. You may have to supplement these classes by specifying the formatting of logical structures peculiar to a given document, such as mathematical formulae. You can modify the standard document classes or create an entirely new one, though one should possess a basic understanding of typographical design before creating a radically new document class.

1.1 Advantages of LaTeX

– The layouts were developed by (American) design professionals, and they produce documents which indeed look as though "they were printed".
– It is extremely simple to compose mathematical formulae.
– The user must only know a few easily memorized commands, which control the logical structure of the document, and (almost) does not have to know the technical details about how a document is formatted.
– More complex components such as footnotes, bibliography, table of contents etc. as well as simple pictures can be produced without great difficulty.
– LaTeX is the "de facto" standard for scientific publications in Europe and the U.S.A., and many publishers (AMS, APS, Elsevier, Springer) promiss a faster turn-around for papers marked up in (pure) LaTeX or in the house style.

1.2 Disadvantages of LaTeX

– TeX is not WYSIWYG, so its use is less intuitive than those systems.
– Rather steep learning curve compared to WYSIWYG systems.
– FrameMaker, Word, etc., offer ever more functionality.
– One can vary a few parameters quite easily within the framework of a given LaTeX document class. However more basic deviations from the available layouts are only possible with a lot of (re)programming.

1.3 How does LaTeX work?

A TeX (LaTeX) input file is a "plain" text file prepared with a text editor and in general has the extension `.tex`.

The files describing a document class (extension `.cls`), a style extension (extension `.sty`) or options (extension `.clo`) are stored in a standard public directory. There exist four standard LaTeX classes, report, article, book and letter. These classes can be customized by specifying one or more options or loading style files.

LaTeX outputs a series of files. One of them (extension `.dvi`) contains a device-independent binary representation of the formatted text. It can be converted into a printable or viewable form by a separate program.

Each (La)TeX run generates a log file (extension `.LIS` or `.log`); it contains information which also appeared on the screen (e.g. the names of the files read, the numbers of the pages processed, error messages etc).

The other files contain information about cross-referencing (extension `.aux`), table of contents (extension `.toc`), list of figures (extension `.lof`) and list of tables (extension `.lot`). They are used in a subsequent LaTeX run to produce particular elements of the document.

A file with extension `.idx` contains raw indexed items. They can be sorted by the `makeindex` program and included as an `.ind` file on a subsequent run.

A file with extension `.bbl` contains bibliographic data extracted by BibTeX from a bibliography database.

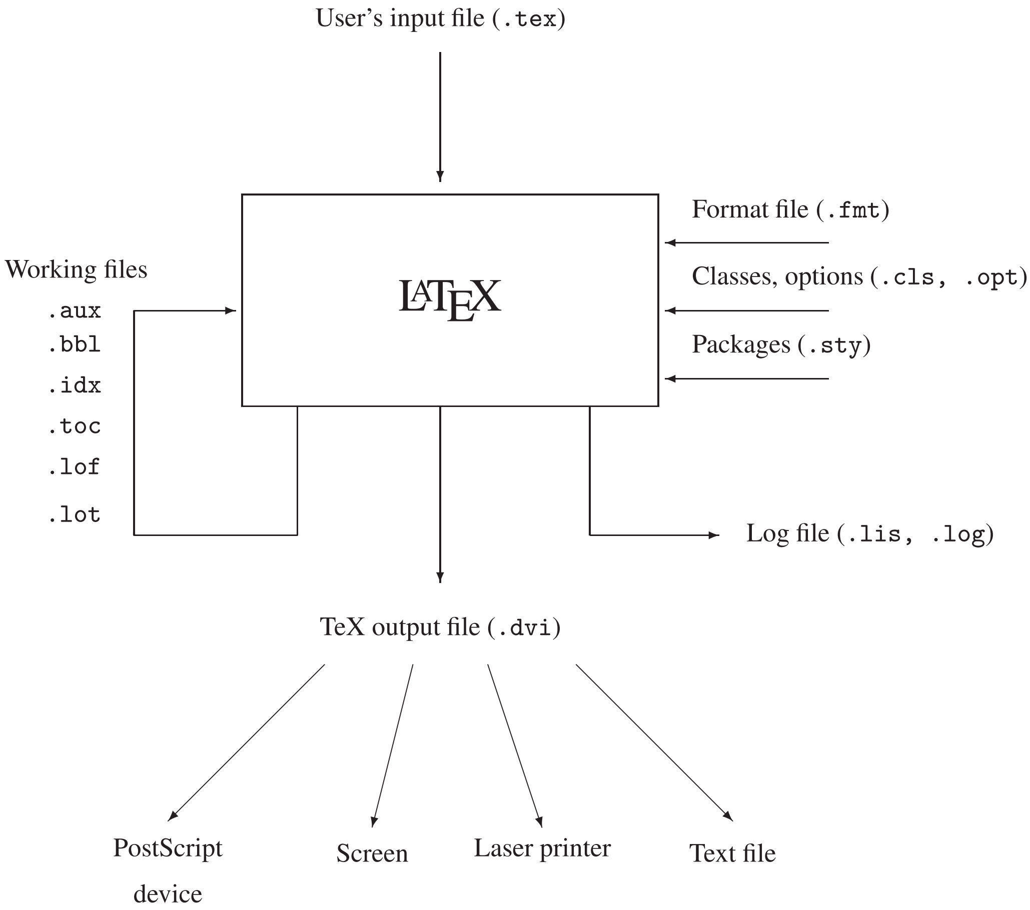


Figure 1.1: Data flow for the files used by LaTeX

1.4 TEX glossary

This section is partly based on a glossary posted to `TEXHAX` by Jacques Goldberg of the Department of Physics, Technion–Israel Institute of Technology, Haifa (Israel).

glyph A graphical representation of a character, for example **a**, *a* and `a` are three different glyphs for the same character.

font A set of glyphs corresponding to a character set.

primitives About three hundred basic commands wired into the TEX program.

macros More sophisticated or simpler to use commands, built upon the primitives.

Plain TEX Donald Knuth's TEX augmented by his own set of macros.

LATEX Donald Knuth's TEX augmented by Leslie Lamport's alternative set of macros.

AMS-TEX Donald Knuth's TEX plus American Mathematical Society macros. The functionality of this package is now available to LATEX users with the `amstex` style option.

format A given set of macros, compiled and written into a file of filetype `fmt` once to save repeated compilation time. TEX will restore all definitions by reading such a format file.

TFM Font Metric files contain *device independent global* information about the space on the output requested for each glyph in the font, as well as other boundary conditions such as relative spacing with neighbouring glyphs (e.g. ligatures). TEX does *not* need to know the detailed shape of a glyph. Filetype is `tfm`.

DVI A DeVice Independent file (extension `dvi`) is output by TEX. DVI files use the ASCII 255-character set and can be copied between different computers (PC's and mainframes) as *binary (image)* files. They contain the description of the formatted document.

DVI driver A program which turns the human-unreadable DVI file into ink on paper or light on a display. The DVI driver *must* know exactly how to produce the image of a given glyph. Therefore it needs a *raster* image for each glyph, and this raster is device (printer, display) dependent, at least, but not only, because devices have different resolutions.

previewer DVI driver to output the DVI file onto a screen.

font file A set of raster representations for each glyph in a font, clearly printer and magnification-dependent. These files have various filetypes such as `pxl`, `pk`, `gf`. The differences in names are historical and technical. `pxl` is obsolete but still used by some old drivers, `pk` is by far the most efficient especially when it comes to saving disk space, `gf` is produced on the way to making fonts by the METAFONT program.
DVI drivers are *not* supposed to be able to magnify fonts, but they should not generate a fatal error when a font is not available. DVI drivers expect to find the magnified rasters in a specific directory structure, and the user is responsible for not invoking magnifications for which no raster files exist; again TEX does *not* check that the driver will or will not be able to print out a font at the requested magnification. This is *not* TEX's business.

magnification A scale factor applied to the original design size for a font. The following magnification factors are accepted:

$$0.5, 0.6, \ldots, 1.0, \sqrt{(}1.2), 1.2, (1.2)^2, etc$$

Not all fonts at each magnification will be present on all machines.

METAFONT A companion program to TEX used to generate new fonts, or existing fonts at new magnifications. The program METAFONT reads a font definition file, actual device parameters, magnification etc., to generate *two* output files per input font file definition. The input is an `mf` file and the two outputs are the corresponding **tfm** and `gf` files. If you wish to make a font at 5 different magnifications you have to run METAFONT 5 times, on almost the same `mf` file. If you wish to

create the font for 3 different devices (a dot matrix printer, a laser printer, a display) you again must run METAFONT 3 times, with the correct printer definition.
All these runs will always produce the same `tfm` *file* but a *different* `gf` *file*.

LaTeX class A file which contains the description of a document instance. These are the files which are specified in the LaTeX command

`\documentclass[options]{class_style}`

where `option` is an optional part, which specifies flags that are recognized inside the code implementing the class (such as `12pt`, `dvips`, `draft`).

LaTeX package A file which contains modifications to a class or additional functions with respect to standard LaTeX. Styles are loaded by the command:

`\usepackage[options]{package_name1,package_name2}`

`options` are flags on which the code inside the packages depends.

1.4.1 Suggested further reading on LaTeX

Many books exist on TeX and LaTeX in almost all languages spoken at CERN. Below, we list only the more general ones. The first four (three in English and one in French) can be bought from the UCO.

- *The TeXbook.* D. Knuth, Addison Wesley, Reading, 1984. ISBN 0-201-13448-9.
- *LaTeX, a Document Preparation System, Second Edition.* L. Lamport, Addison Wesley, Reading, 1994. ISBN 0-201-52983-1.
- *The LaTeX Companion.* M. Goossens, F. Mittelbach and A. Samarin, Addison Wesley, Reading, 1994. ISBN 0-201-54199-8.
- *LaTeX guide pratique, édition revue et augmentée.* (in French) Christian Rolland, Addison Wesley France, Paris, 1995. ISBN 2-87908-104-1.
- *Math into LaTeX. An Introduction to LaTeX and AMS-LaTeX.* George Grätzer, Birkhäuser, Boston, 1996. ISBN 0-8176-3805-9.
- *LaTeX, Band 1, Einfürung.* (in German) H. Kopka Addison Wesley Deutschland, Bonn, 1994. ISBN 3-89319-664-1.
- *LaTeX, Band 2, Ergänzungen — mit einer Einfürung in METAFONT.* (in German) H. Kopka Addison Wesley Deutschland, Bonn, 1995. ISBN 3-89319-665-1.
- *LaTeX, Band 3, Einweiterungen.* (in German) H. Kopka Addison Wesley Deutschland, Bonn, 1997. ISBN 3-89319-666-1.

1.5 Running LaTeX at CERN

Recent version of LaTeX, TeX, METAFONT and their companion programs have been installed on all centrally supported computer platforms at CERN. Below we show the basic commands to format a LaTeX document, transform the DVI file into PostScript and print the output. Then each major system will is reviewed in turn.

1.5.1 Running a Sample File

Before preparing your own documents, you may want to get acquainted with LaTeX by running it on a sample input file. You can, for instance, copy the file `sample2e.tex` to your working directory, although this is not strictly necessary, since TeX is set up in a way that it will find that file automatically.
On VMS, the file is in `TEX_DISK:[TEX.NEWTEX.LATEX2E]SAMPLE2E.TEX` and on Unix in the usual TeX tree `/usr/local/share/texmf/tex/latex/base/sample2e.tex`.
You can now modify your copy of the file `sample2e.tex` if you want and then run LaTeX on your file by typing (VMS Users, do not forget to initialize the TeX system variables by typing the command `cerntex` once at the beginning of your session):

```
latex sample2e
```

When LaTeX has finished, it will have produced the file `sample2e.dvi` in your current (working) directory. You can print this file by first transforming the information into PostScript using the command `dvips`.

```
dvips sample2e.dvi
```

In general the extension `.dvi` can be omitted. By default, at CERN `dvips` is configured to save the output into a file, which can then be sent separately to a PostScript printer. On Unix, the DVI-to-PostScript conversion program `dvips`, can be easily customised to suit user preferences. It is enough to deposit a file called `.dvipsrc` in your home directory. This files specifies runtime options for `dvips`. For example, a file containing the line:

```
o | xprint
```

instructs `dvips` to pipe the PostScript output it generates directly into `xprint` (without writing a persistent PostScript output file to your current directory).

```
XPRINT/LW/PR=printer_name SAMPLE2E.PS       -- on VMS
xprint -P printer_name sample2e.ps          -- on Unix
```

After your output has been printed, you can delete the files `sample2e.dvi` and `sample2e.ps`.

1.5.2 Documentation

Documentation is stored on afs in the directory `/usr/local/share/texmf/doc` or `/usr/local/doc/TeX`. It is available conveniently on the Web at the URL `http://wwwcn.cern.ch/dci/asis/products/TeX/index.html`.

1.6 TEX on VMS

The version of TEX installed on VMS is based on the DECUS `TEX91` tape collection. To set up the necessary TEX environment variables and symbols, all TEX users must execute the command CERNTEX or enter it in their `LOGIN.COM`.

1.6.1 Interrupting TEX

When TEX is running you can interrupt it by typing `control-C`. This will stop TEX as if it had encountered an ordinary error, and you can then return to `DCL` command level by typing `X`, as described in the TEXbook. Note, that the program closes the output files (`.dvi`, `.lis`, etc). Of course, `control-Y` will also stop execution, but the `.dvi` file will be corrupted.

A "deadlock" may occur if LaTeX requires a file which is not accessible. The user will be prompted to input the correct file name, if any. If you want to exit from TEX at that point, type `control-Z` or specify `null` as filename. This is an empty file, which is provided in the standard input file directories.

1.6.2 `xdvi` – DVI previewer for DECWindows

The DVI previewer `xdvi` is a program running under DECwindows. It has the capability of showing the file shrunken by various (integer) factors, and also has a "magnifying glass" which allows one to see a small part of the unshrunk image momentarily.

Before displaying any page or part thereof, it checks to see if the DVI file has changed since the last time it was displayed. If this is the case, then `xdvi` will reinitialize itself for the new DVI file. For this reason, exposing parts of the xdvi window while TEX is running should be avoided. This feature allows you to preview many versions of the same file while running `xdvi` only once.

```
   xdvi [+[<page>]] [-s <shrink>] [-density <%>] [-p <pixels>] [-l] [-rv]
[-paper <papertype>] [-mgs[n] <size>] [-altfont <font>]
[-margins <dimen>] [-sidemargin <dimen>] [-topmargin <dimen>]
[-offsets <dimen>] [-xoffset <dimen>] [-yoffset <dimen>] [-keep]
[-hushspecials] [-hushchars] [-hush]
[-fg <color>] [-bg <color>] [-hl <color>] [-bd <color>] [-cr <color>]
[-bw <width>] [-geometry <geometry>] [-icongeometry <geometry>]
[-iconic] [-display <host::display>] [-copy] [-thorough] dvi_file
```

The options, keystrokes and mouse actions of `xdvi` are described in the manual page obtained by typing `help xdvi`.

1.6.3 `dvips` - DVI convertor to PostScript

The DVI driver `dvips` translates a DVI file into a PostScript file. The latter can be printed on any PostScript printer. The syntax for the `dvips` command is

```
DVIPS options file[.dvi]
```

The output file is created in the default directory with the same filename as the DVI file and with extension `PS`. Possible options are:

`/OUTPUT=filename`	The output file will be called `filename`.
`/NUMBER=num`	At most num pages will be printed. Default is 100000.
`/START=num`	First page printed is the one numbered `num`. Default is first page in the DVI file.

Other options of `DVIPS` can be found by issuing the command `HELP DVIPS`.

1.6.4 `dvi2tty` – DVI previewer for on ordinary terminals

The `DVI2TTY` driver converts a DVI file to a format that is appropriate for terminals and line printers. The program is intended to be used for preliminary proofreading of TeX-ed documents. By default the output is directed to the terminal, possibly through a pager (depending on how the program was installed), but it can be directed to a file. The command format is:

```
DVI2TTY [options] file[.dvi]
```

The most important options (typing `dvi2tty` without arguments will show all available options) are:

`-ofile`	Write output to file `file`.
`-plist`	Print the pages chosen by list.
`-wn`	Specify terminal width. Default is 80.

1.7 TeX on Unix

1.7.1 Introduction

The version of the TeX installed on Unix is `teTeX` 0.4, based on Karl Berry's `web2c 6.1` sources and the Knuth's TeX engine 3.14159. It is the summer 1996 release of LaTeX which is installed (formally known as `LaTeX2e <1996/06/01>`). It is based on the TDS (TeX Directory Structure) setup and follows the GNU convention to store the machine-independent files in the directory `/usr/local/share` rather than `/usr/local/lib`. The vast majority of the files comes straight from the `teTeX` distribution. Only a few CERN-specific extensions have been added. It is anticipated that the standard part will be regularly updated, while the CERN extensions will be kept unaltered as far as possible

At present this set-up is available only on UNIX (including Linux). As soon as the same functionality (including an efficient recursive directory searching strategy) becomes available on VMS, Windows 95/NT or Mac, it will be installed immediately.

Pure LaTeX 2$_\varepsilon$ Behaviour

The latest version of LaTeX 2$_\varepsilon$ no longer permits the mixing of the LaTeX 2.09 declaration `\documentstyle` and the LaTeX 2$_\varepsilon$ `\usepackage` command. You should use the pure LaTeX 2$_\varepsilon$ syntax `\documentclass`.

LaTeX 2.09 CERN styles no longer available

The only supported CERN classes are those compatible with LaTeX 2$_\varepsilon$. For instance, the style `cernlett.sty` is no longer available, and is replaced by `cernlett.cls`. This means that when in the past you wrote:

```
\documentstyle[cernlett]{letter}
```

you now simply specify:

```
\documentclass{cernlett}
```

1.7.2 New "lighter" fonts

Knuth's Computer Modern (CM) fonts have been developed for use on high resolution typesetters and are characterised by quite thin stems (i.e., they seem quite "light" with respect to Times or Helvetica). As Computer Modern fonts are, by default, loaded as bitmap images generated for a particular printer from the METAFONT sources, we had previously optimised the bitmaps for the 300 dots-per-inch (dpi) LaserWriter, which was, until recently, the most widely used printer at CERN. However, today, most printers have a resolution of at least 600 dpi, and we have installed fonts which are optimised for the Hewlett Packard LaserJet printers which are now standard at CERN. Nevertheless, as many users at CERN grew accustomed to the lower precision "fat" versions, I have installed on the central systems (CERNSP and HPPLUS) a script `fatfonts.sh`, which will load the 300 dpi fatter version of the Computer Modern fonts in your document.
As an alternative to using specific bitmapped versions of the Computer Modern fonts, you can load the publicly available PostScript Type 1 renderings developed by our colleague Basil Malyshev of IHEP (Protvino, Russia). These are included in your document by the command:

```
dvips -Pbkm filename
```

1.7.3 Multiple Languages and the `babel` Package

A whole set of LaTeX formats have been generated where, besides hyphenation patterns for English, also those for a second language are included. The default `latex` command will run a version of TeX which contains English and German hyphenation patterns. The table below shows the two-letter code you have to prepend to the `latex` command to gain access to a given language-pair combination.

`ca` Catalan	`cz` Czech	`de` German	`dk` Danish	`es` Spanish	`fi` Finnish	`fr` French	`bg` G.B. English
`hr` Croatian	`hu` Hungarian	`it` Italian	`nl` Dutch	`no` Norwegian	`pl` Polish	`sk` Slovak	`sv` Swedish

Thus, the command `frlatex` will load LaTeX with the French (and English) hyphenation patterns, `itlatex` with Italian and English, etc. Also, the preamble part of a typical job using these languages should look something like (in the case of French):

```
\documentclass{article}
\usepackage[francais]{babel}
%%%% NOT \usepackage[french]{babel}! %%%%%
...
\begin{document}
```

A word of caution for French, namely for French the following declaration:

```
\usepackage[french]{babel}
```

should at present not be used, since the `french` package of Bernard Gaulle is not correctly interfaced to the standard version of Babel (3.5) which we are running at present. This will be fixed in the next release.

1.7.4 Searching supplementary Directories

If you, your Experiment, Group, University, etc. have a repository containing supplementary TeX-related files, and you would like LaTeX, dvips, etc. to also automatically search that directory (tree), then you should append a pointer to that directory to the relevant TeX environment variables. For instance, in the case of the `TEXINPUTS` variable, which controls the search for input files by (La)TeX, you could add the new directory newdir to *the end* of TeX's search path as follows:

```
TEXINPUTS=:newdir:
export TEXINPUTS          # bash, sh, ksh

setenv TEXINPUTS :newdir  # csh, tsh
```

This will first search the standard directory tree, and only then the specified directory. If you point to the top of a directory *tree*, and you want the complete hierarchy to be searched, then you should append a double slash to the end of the name, for instance

```
TEXINPUTS=:topdir//
export TEXINPUTS            # bash, sh, ksh

setenv TEXINPUTS :topdir//  # csh, tsh
```

In this case TeX will scan `topdir` and all its subdirectories to resolve undefined file references.

1.7.5 Customising the `dvips` Program

The DVI-to-PostScript conversion program `dvips`, which is used at CERN to take the output of TeX and transform it into (Encapsulated) PostScript, can be easily customised to suit your preferences. It is enough to deposit a file called `.dvipsrc` in your home directory. This files specifies runtime options for `dvips`. For example, a file containing the line:

```
o | xprint
```

instructs `dvips` to pipe the PostScript output it generates directly into `xprint` (without writing a persistent PostScript output file to your current directory, as is now the default at CERN).

1.7.6 Handling Encapsulated PostScript Files in LaTeX

Almost every day LaTeX users come to see me because they have problems including PostScript files in their LaTeX documents. And, almost invariably, the problem is due to the fact that they are dealing with PostScript files, which are generated for a printer, and not encapsulated for inclusion inside a document. Most printer drivers on MS Windows, Apple, or UNIX will add all kinds of printer-specific commands into the PostScript file, thus upsetting the encapsulation. In fact, the Adobe standard is quite explicit about this (Appendix H of the PostScript Reference Manual).

Thus, every time you use a graphics tool for generating images that you want to include in a LaTeX (or Word, FrameMaker, PageMaker, etc.) document, make sure that you save it as an *Encapsulated PostScript* (EPS) file. Many public domain and commercial tools and programs, like HIGZ, `convert` or Adobe Illustrator, Coral Draw on Macs and PC's can generate EPS files.

If you have a PostScript file and you cannot regenerate it easily as a *bona fide* EPS file, then you could try to run it through Adobe Distiller. This program will turn your PostScript file into the PDF (Portable Document Format) format, which can be read into Adobe Illustrator, and then saved as an EPS file. This means some work, but the investment in time will allow you to minimize probable problems downstream.

1.7.7 Preparing Articles for Springer Verlag

Springer Verlag is a publisher that publishes (amongst many other journals) "*Zeitschrift für Physik*". However, like the APS with REVTeX, they still use a version of LATEX style files which work with a non-standard old set-up, so special formats had to be created at CERN to allow users to prepare articles for submission to Springer journals. This procedure is available on both the CERNSP and HPPLUS central systems, and is called `springer.sh`. It is a shell script with two parameters: the first is the format (`cljour2` for Computer Modern fonts and `pljour2` for PostScript fonts) and the second is the name of your file. A sample run with a demonstration file for the journal "*Zeitschrift für Physik*" is shown below. You can try this on the central systems yourself (note that you will have to run the script twice to resolve the cross-references).

```
> springer.sh pljour2 zphysc.dem
This is TeX, Version 3.14159 (C version 6.1)
(/afs/cern.ch/user/g/goossens/public/zphys/zphysc.dem
LaTeX Version 2.09 <25 March 1992>
This is pljour2.
(zphysc.aux) [59]
390 author names and
36 institutes processed
[60] [60] [61] [62] [63]
Underfull \vbox (badness 10000) has occurred while \ouput is active [64]
Underfull \vbox (badness 2088) has occurred while \output is active [65]
[66] (zphysc.aux) )
(see the transcript file for additional information)
Output written on zphysc.dvi (9 pages, 64536 bytes).
Transcript written on zphysc.log.
```

1.7.8 Submitting Electronic Copies to the Publishers

Many publishers now accept electronic submissions of articles, and promise a faster turnaround if they do not have to (re)type your article themselves. However, some of these publishers just provide a page or two of printed instructions about which layout should be used, and leave it up to the authors to write the necessary LATEX code to implement their design. As most CERN authors are not TEX gurus, they are usually unable to develop themselves the necessary class file.

As I have stated several time before, it is up to the publishers to provide the tools to submit articles to the journals they manage. It is they who benefit financially and otherwise from electronic submission, so it is only logical that they should invest the effort to develop the necessary support files to implement their house styles. It cannot be considered CERN's task to write such files, since not only would this be an extremely inefficient duplication of effort, but also no resources are available in this area.

Another point to watch is that most publishers allow you to use only *vanilla-flavoured LATEX*, possibly augmented with a few of their journal-specific extensions. Just as an example, the REVTeX family of journals comes with the following explicit instructions:

The use of author-created macros is strongly discouraged. The REVTeX macro set is intended to represent completely the tagging needs for a physics manuscript. Non-standard tags will probably disqualify a manuscript file from a compuscript program because most journal compositors cannot translate them cost-efficiently.

The time and effort required to massage compuscript files that do not follow these standard formatting guidelines will eliminate any cost savings and/or time savings in production, and will eliminate any proofreading time savings for the author.

Therefore, authors who use their own custom macros to save keystrokes in preparing their manuscript should remove them before generating the manuscript output for submission.

1.7.9 What about your Legacy Files

As a special convenience for those of you having old (pre-1992) files for which you cannot invest the necessary time to update them to the standard version of LaTeX 2_ε, I have made available *on CERNSP only* the shell script `oldlatex.sh`, which will run a document with the old LaTeX set-up. Note that there is no guarantee that your old files will still run or be printable even with this set-up. However, several tests have shown that *well-behaved* old files compile and print without problems.

1.7.10 Where are those TeX files anyway?

The `teTeX` distribution comes with a set of tools to manage the TeX file base. The more useful is the command `kpsewhich`, which has the following usage pattern:

```
kpsewhich: [options] pathtype filename
```

Valid options are the following:

`-n progname`	pretend to be progname to `kpathsea`;	`-p`	act like `kpsepath`
`-m mode`	set METAFONT mode;		
`-w`	act like `kpsewhich`	`-v`	act like `kpsexpand`

Valid values of `pathtype` are

`gf`	generic font bitmap	`mppool`	METAPOST program strings
`pk`	packed bitmap font	`mpsupport`	METAPOST support files
`base`	METAFONT memory dump	`pict`	Other kinds of figures
`bib`	BIBTeX bibliography source	`tex`	TeX source
`bst`	BIBTeX style files	`texpool`	TeX program strings
`cnf`	`kpathsea` runtime configuration files	`tfm`	TeX font metrics
`fmt`	TeX memory dump	`vf`	virtual font
`mem`	METAPOST memory dump	`dvips_config`	`dvips` config files
`mf`	METAFONT source	`dvips_header`	`dvips` header files
`mfpool`	METAFONT program strings	`troff_font`	`troff` fonts
`mp`	METAPOST source		

Examples of use are the following:

```
> kpsewhich -n tex tex array.sty
/usr/local/share/texmf/tex/latex/tools/array.sty
> kpsewhich -n dvips dvips_config config.bkm
/usr/local/share/texmf/dvips/bakoma/config.bkm
```

The first command finds the `array` style in LaTeX's `tools` directory, while the second looks for the `dvips` configuration file `config.bkm`, which contains the necessary information to use Malyshev's PostScript instances of the CM fonts (see above).

1.7.11 When in trouble

When you get a `latex: not found` error from the system, first check whether the above directory is in your command search path by typing `echo $PATH`. If it is, please type `ls /usr/local/bin/latex` to check whether the `latex` file is indeed installed. If it is not, contact your system administrator or, in his absence, Philippe Defert/IT for more information about getting it up and running on your own Unix system. The installation of software from the asis server is described in [14].

Once you get LaTeX started, you should get the message:

```
This is TeX, Version 3.14159 (C version 6.1)
LaTeX2e <1996/06/01>
Hyphenation patterns for english, german, loaded.
```

If the command `latex` is found, but you do not get this far, then you have probably the wrong version of the LaTeX format. In this case you should contact the people mentioned above.

1.7.12 Interrupting TEX

When TEX is running you can interrupt it by typing `control-C`. This will stop TEX as if it had encountered an ordinary error, and you can then return to Unix shell command level by typing `X`, as described in the TEXbook.

A "deadlock" may occur if LaTeX requires a file which is not accessible. The user will be prompted to input the correct file name, if any. If you want to exit from TEX at that point, specify `null` as filename. This is an empty file, which is provided in the standard input file directories. You should be aware that in some cases TEX only searches its own defaults input directories. Also when a file to be input is not in the same directory as the root-file, *absolute pathnames* have to be used to transmit the file's actual pathname to TEX. In particular the ~ character for the home directory is not recognized.

1.7.13 `dvips` - DVI convertor to PostScript

The DVI driver `dvips` translates a DVI file into a PostScript file. The latter can be printed on any PostScript printer, usually an Apple LaserWriter.

The syntax for the `dvips` command is:

```
This is dvipsk 5.58f Copyright 1986, 1994 Radical Eye Software
     Usage: dvips [options] filename[.dvi]
 a*  Conserve memory, not time      y # Multiply by dvi magnification
 b # Page copies, for posters e.g.  A   Print only odd (TeX) pages
 c # Uncollated copies              B   Print only even (TeX) pages
 d # Debugging                      C # Collated copies
 e # Maxdrift value                 D # Resolution
 f*  Run as filter                  E*  Try to create EPSF
 h f Add header file                F*  Send control-D at end
 i*  Separate file per section      K*  Pull comments from inclusions
 k*  Print crop marks               M*  Don't make fonts
 l # Last page                      N*  No structured comments
 m*  Manual feed                    O c Set/change paper offset
 n # Maximum number of pages        P s Load config.$s
 o f Output file                    R   Run securely
 p # First page                     S # Max section size in pages
 q*  Run quietly                    T c Specify desired page size
 r*  Reverse order of pages         U*  Disable string param trick
```

```
s*  Enclose output in save/restore V*  Send downloadable PS fonts as PK
t s Paper format                  X # Horizontal resolution
x # Override dvi magnification    Y # Vertical resolution
                                  Z*  Compress bitmap fonts
-    Interactive query of options
pp   #-# First-last page
mode modename

    # = number   f = file   s = string  * = suffix, '0' to turn off
    c = comma-separated dimension pair (e.g., 3.2in,-32.1cm)
```

Fonts used may either be resident in the printer or defined as bitmaps in PK files, or a "virtual" combination of both. dvips will automatically invoke METAFONT to generate non-available fonts via the MakeTeXPK script.

1.7.14 xdvi – DVI previewer for X-windows

The DVI previewer xdvi is a program which runs under the X window system. It has the capability of showing the file shrunken by various (integer) factors, and also has a "magnifying glass" which allows one to see a small part of the unshrunk image momentarily.

Before displaying any page or part thereof, it checks to see if the DVI file has changed since the last time it was displayed. If this is the case, then xdvi will reinitialize itself for the new DVI file. For this reason, exposing parts of the xdvi window while TeX is running should be avoided. This feature allows you to preview many versions of the same file while running xdvi only once.

In addition to using keystrokes to move within the file, xdvi provides buttons on the right side of the window, which are synonymous with various sequences of keystrokes.

```
> xdvi -help
xdvik version 18f
Usage: xdvi [+[<page>]] [-d <debugnum>] [-s <shrink>] [-S <density>] [-nogrey]
        [-gamma <g>] [-p <pixels>] [-margins <dimen>] [-sidemargin <dimen>]
        [-topmargin <dimen>] [-offsets <dimen>] [-xoffset <dimen>]
        [-yoffset <dimen>] [-paper <papertype>] [-altfont <font>] [-l] [-rv]
        [-expert] [-mgs[n] <size>] [-hush] [-hushspecials] [-hushchars]
        [-hushchecksums] [-fg <color>] [-bg <color>] [-hl <color>]
        [-bd <color>] [-cr <color>] [-bw <width>] [-display <host:display>]
        [-geometry <geometry>] [-icongeometry <geometry>] [-iconic] [-keep]
        [-copy] [-thorough] [-nopostscript] [-noghostscript] [-version]
        [-maketexpk] [-mfmode <string>] [dvi_file]
```

The many options of xdvi are described in the manual page obtained by typing man xdvi.

1.7.15 gs – a PostScript previewer

The public domain Ghostscript previewer gs is available on Unix workstations. It is part of the GNU project. Ghostscript is a programming language similar to Adobe Systems' PostScript (tm) language. Gs reads a file and displays it as a Ghostscript file. It then interprets commands from standard input until an end-of-file character (control-D) is typed. The 'quit' character (control-C) also terminates Ghostscript execution.

```
gs -help
Aladdin Ghostscript 4.03 (1996-9-23)
Copyright (C) 1996 Aladdin Enterprises, Menlo Park, CA.  All rights reserved.
Usage: gs [switches] [file1.ps file2.ps ...]
```

```
Most frequently used switches: (you can use # in place of =)
 -dNOPAUSE           no pause after page   | -q       'quiet', fewer messages
 -g<width>x<height>  page size in pixels   | -r<res>  pixels/inch resolution
 -sDEVICE=<devname>  select device         | -c quit  (as the last switch)
                                           |            exit after last file
 -sOutputFile=<file> select output file: - for stdout, |command for pipe,
                                         embed %d or %ld for page #
Input formats: PostScript PostScriptLevel1 PostScriptLevel2 PDF
Available devices:
   x11 x11alpha x11cmyk x11mono deskjet djet500 laserjet ljetplus ljet2p
   ljet3 ljet4 cdeskjet cdjcolor cdjmono cdj550 pj pjxl pjxl300 bj10e bj200
   bjc600 bjc800 faxg3 faxg32d faxg4 pcxmono pcxgray pcx16 pcx256 pcx24b pbm
   pbmraw pgm pgmraw pgnm pgnmraw pnm pnmraw ppm ppmraw tiffcrle tiffg3
   tiffg32d tiffg4 tifflzw tiffpack tiff12nc tiff24nc psmono bit bitrgb
   bitcmyk pngmono pnggray png16 png256 png16m pdfwrite nullpage
Search path:
   . : /usr/local/share/ghostscript/4.02 :
   /usr/local/share/ghostscript/fonts
For more information, see /usr/local/share/ghostscript/4.02/doc/use.txt.
Report bugs to ghost@aladdin.com; use the form in new-user.txt.
```

The program `ghostview` provides a menu/mouse interface for the `ghostscript` previewer. There is also a more modern variant gv To invoke this interface, give the command

```
ghostview  <filename>
```

1.8 TEX on personal computers

1.8.1 TEX on the PC

For Windows 95 or NT there is at present no good public domain version. There are a few commercial products, but they are quite expensive.

1.8.2 TEX on Macintosh

Similarly on Mac there are a few shareware versions (like `oztex` and `cmactex`). CERN has decided to use the nicely integrated commercial MacIntosh implementation TEXtures v1.7, and has negotiated with Blue Sky Research, the vendor of TEXtures v1.7, a site licence for 300CHF per copy (instead of the full price of close to 1000 CHF). Those interested should contact David Stungo (AS/MI).

Chapter 2: An Overview of LaTeX2e

2.1 LaTeX 2$_\varepsilon$, Basic Commands

This article gives an overview of the new or extended user commands available with LaTeX 2$_\varepsilon$, the new LaTeX release, compared to the previous version LaTeX 2.09. After introducing the new preamble commands, the extensions for defining new commands and environments, and handling length and boxes are discussed. The new font selection commands are explained, both for text and math, and it is shown how to easily use different font families. A list of supported class and package files is given and new possibilities for controlling page contents and floats are discussed. Most of this material is described in much greater detail in "The LaTeX Companion" [2] and in the second edition of the LaTeX Reference Manual [1][1].

2.1.1 Why LaTeX 2$_\varepsilon$?

Ever since LaTeX became generally available in 1986, its popularity has increased, and many extensions have been developed. Unfortunatlely, these extensions were included in incompatible formats, e.g., "standard" LaTeX with and without NFSS, SLiTeX, $\mathcal{A}\mathcal{M}\mathcal{S}$-LaTeX, and so on. From the LaTeX source alone, it was difficult to determine for which of these (or other) formats a document was composed and, because different sites could have different configuration, document portability was a problem.

Already in 1989 at the Stanford TUG Conference, Frank Mittelbach and Rainer Schöpf got together with Leslie Lamport to discuss these (and other) topics and they published their ideas about possible ways to evolve LaTeX in TUGBoat [3, 4]. This led a few years later to the start of the long-term LaTeX3 project [5–8].

However, to help end the confusion for the present LaTeX users, after a meeting in Spring 1993 between Leslie Lamport and Frank Mittelbach in Mainz, it was decided to release an upgraded version of LaTeX, called LaTeX 2$_\varepsilon$, which was officially announced in August 1993 at the TUG Conference at Aston.

Its stated aims are:

- bring all extensions back under a single format;
- prevent proliferation of mutually incompatible dialects of LaTeX 2.09;
- NFSS becomes the "standard" font selection scheme;
- make style files like amstex (formerly $\mathcal{A}\mathcal{M}\mathcal{S}$-LaTeX format) or slides (formerly SLiTeX format) into extension packages, all using the same base format;
- add a small number of often-requested features;
- retain the "touch and feel", or the "flavour" of LaTeX 2.09.

The first production version of LaTeX 2$_\varepsilon$ was released on June 1st 1994. Twice a year (in June and in December) consolidation releases are planned, in order to keep all versions of the files in synchronization. Intermediate bugs are corrected via patch files. Bug reports are handled centrally by inviting the users to fill out an electronic form, distributed with the LaTeX 2$_\varepsilon$ distribution, and sending it via electronic mail to `latex-_bugs@rus.uni-_stuttgart.de`. Note that only bug reports referring to the last two releases will be considered.

2.1.2 Initial and preamble commands

In this section commands which can only be used before or in the preamble will be discussed. The first two below can only appear *before* the `\documentclass` command.

[1]Note that these two books are complementary, with the first for users who have already some experience with LaTeX, like most people at CERN, while the second is the "official" user guide for the language. Both can be purchased from the UCO by using the EDH system. Send mail to `User.Support@cern.ch` indicating which book(s) you want, the budget code, and the identity of the budget holder. A complete list of books sold at the UCO is available at the URL `http://consult.cern.ch/books`

Initial commands

`\NeedsTeXFormat{`*format-name*`}[`*release-date*`]`

This command, which is normally present in package and class files, can also be useful in user documents to make sure that the file is run with LaTeX 2$_\varepsilon$. Users who try to run it with LaTeX 2.09 or plain TeX will get a reasonably clear error message. An example is

```
\NeedsTeXFormat{LaTeX2e}[1994/12/01]
```

If you want to make sure that your document can be processed at another site, it could make sense to include all packages and files that your document needs together with the main file. LaTeX 2$_\varepsilon$ provides the following syntax to facilitate this

```
\begin{filecontents}{file-name}
    file-contents
\end{filecontents}
```

When your document file is run through LaTeX 2$_\varepsilon$ the body of each `filecontents` environment is written verbatim to a file whose name is given as the argument *file-name*. If a file with such a name already exists in any of the directories "visible" to TeX, only an informative message is given, the body of the environment is by-passed, and the file is not replaced.

Preamble commands

The next commands in the preamble are specifically designed to differentiate LaTeX 2$_\varepsilon$ documents from those needing LaTeX 2.09.

`\documentclass[`*option-list*`]{`*class-name*`}[`*release-date*`]`

This command or "declaration" replaces the LaTeX 2.09 command `\documentstyle`.

There must be exactly one `\documentclass` declaration in a document, and it must come first (except for the "initial" commands described above).

option-list: list of options that each can modify the formatting of document elements defined in the *class-name* file or in packages loaded with `\usepackage` declarations, as described below.

class-name: name of the class file (file extension `.cls`).

release-date (optional) specifies release date of the class file, using the format `YYYY/MM/DD`. If a version of the class older than this date is found, a warning is issued.

`\documentstyle[`*option-list*`]{`*style-name*`}[`*release-date*`]`

This command, which is supported for compatibility reasons, is similar to `\documentclass`, but it loads a "compatibility mode" which redefines certain commands to act as they did in LaTeX 2.09 and thus allows you to run your old files unchanged through LaTeX 2$_\varepsilon$. Note, however, that in this mode, you *cannot* use any of the LaTeX 2$_\varepsilon$ extensions described in this article.

You can define new or change existing document elements by loading package files with `\usepackage`, whose syntax is:

`\usepackage[`*option-list*`]{`*package-name*`}[`*release-date*`]`

package-name: name of the package (file extension `.sty`); a package can

– define new document elements;
– modify elements defined in the class file;
– extend the range of documents that can be processed.

option-list: list of options, each of which can modify the formatting of elements defined in the package.

release-date: (optional) earliest desired release date of package file (see `\documentclass` command above).

Any number of `\usepackage` are allowed, but LaTeX 2_ε makes sure that each package is loaded only once. On top of processing the list of options specified in the argument *option-list* on its `\usepackage` command, each package also processes the option *option-list* on the `\documentclass` command.

`\listfiles`

To help you get an overview of the files read in by your document during processing, you can place a `\listfiles` command in the preamble of your document. This will display the list of all files used at the end of the run.

Example of document preambles

The following preamble references the article class with the (global) options twocolumn and a4paper, and loads the multicol and babel packages, the latter with the german and french options. Other document parameters (e.g., the `textheight`) can also be specified.

```
\documentclass[twocolumn,a4paper]{article}
\usepackage{multicol}
\usepackage[german,french]{babel}
\addtolength{\textheight}{2cm}
\begin{document}
    ...
\end{document}
```

The following shows three equivalent ways of specifying the loading of packages.

```
\documentclass[german]{book}
\usepackage[german]{babel}
\usepackage[german]{varioref}
\usepackage{multicol}
\usepackage{epic}
```

Somewhat less verbose is:

```
\documentclass[german]{book}
\usepackage[german]{babel,varioref}
\usepackage{multicol,epic}
```

With german as global option you can write:

```
\documentclass[a4paper,german]{book}
\usepackage{babel,varioref,multicol,epic}
```

A complex document might look something like the following:

```
\NeedsTeXFormat{LaTeX2e}[1994/05/01]
\begin{filecontents}{varioref.sty}
   ....          % Code for varioref package
\end{filecontents}
\listfiles       % print list of files referenced
\documentclass[a4paper,german]{book} % book class
\usepackage{varioref}
\begin{document}
%--------------------- front matter of document
\maketitle
\section*{...}    % e.g. section named "Preface"
\tableofcontents  % chapter with table of contents
\listoffigures    % chapter with list of figures
\listoftables     % chapter with list of tables
%--------------------- body of the document
\part{...}
\chapter{...}
   \section{...}
\chapter{...}
\part{...}
%--------------------- back matter of document
\appendix
\chapter{...}     % chapters labelled appendix
\chapter{...}
\begin{thebibliography}
  ...             % bibliography entries
\end{thebibliography}
\begin{theindex}
  ...             % index entries
\end{theindex}
\end{document}
```

Note that, to ensure that the recipient of the document can process the file correctly, the code of the varioref package is shipped with the file inside a `filecontents` environment.

2.1.3 Option processing

Options that are specified in the *option-list* argument of the `\documentclass` or `\usepackage` commands are handled as follows:

1. They are first divided into two types, *local* and *global*:
   - for a *class*, the options from its `\documentclass` command are local and there are no global options;
   - for a *package*, the options from its `\usepackage` command are local but the options from the `\documentclass` command are global.
2. The local and global options that have been declared within the current class or package are processed first, normally in their order of declarations, thus their order in *option-list* is irrelevant.
3. Any local options not declared in the current class or package are then processed. For document classes, this usually means that they are ignored, except for this fact being recorded by adding the option to a list of "unused options"; they may, of course, be used later since they become global options for every package subsequently loaded. For packages, usually an error message is produced, giving the choice of retyping the option name in case it is incorrect.

Finally, when the `\begin{document}` command is reached LaTeX 2_ε will produce a list of all global options not used by the class or any package file, and issue a warning message for each.

2.1.4 Defining new commands and environments

This section and the following describe commands and environments that are used inside the document body (i.e., after the `\begin{document}`) command. Let us first look at what is available for defining new commands and environments.

Defining commands

Commands are defined or redefined in LaTeX with:

```
\newcommand{\mycom}[narg][default]{command definition}
\renewcommand{\mycom}[narg][default]{command definition}
\providecommand{\mycom}[narg][default]{command definition}
```

The first and second commands show enhancements with respect to LaTeX 2.09 by providing the possibility to have an *optional* argument when (re)defining a user command. The presence of such an optional parameter is flagged by the presence of the second optional argument *default*, which specifies the default value of that argument when it is not specified explicitly when the command is used. The last form is useful for general purpose files that are included in a document and over which the user does not always have control (e.g., BIBTeX databases). If `\mycom` is not yet defined, the `\providecommand` will act as `\newcommand` and define it, otherwise the existing definition is kept.

The number of arguments, which *includes* the optional argument, is in the range 0≤*narg*≤9. If the command has no arguments, then the `[0]` can be omitted. Inside the *command definition* part, the arguments are referenced as `#1` to `#`*narg*, the optional argument, if present, being the first one.

For example, compare the following commands, with no, one mandatory, one optional, and one optional and one mandatory argument, allowing the user more freedom in each case.

```
\newcommand{\seq}{x_{0},\ldots\,x_{n}}
\newcommand{\seqm}[1]{#1_{0},\ldots\,#1_{n}}
\newcommand{\seqo}[1][k]{x_{0},\ldots\,x_{#1}}
\newcommand{\seqom}[2][k]{#2_{0},\ldots\,#2_{#1}}
\[\seq\quad\seqm{z}          \]
\[\seqo\quad\seqo[l]         \]
\[\seqom{y}\quad\seqom[i]{q}\]
```

$$x_0,\ldots\,x_n\quad z_0,\ldots\,z_n$$

$$x_0,\ldots\,x_k\quad x_0,\ldots\,x_l$$

$$y_0,\ldots\,y_k\quad q_0,\ldots\,q_i$$

If a command should work both in math and in text mode, special care should be taken in its definition. In LaTeX 2_ε you have the following command:

```
\ensuremath{math code}
```

As its name implies `\ensuremath` ensures that its argument is always typeset in math mode by surrounding it if necessary with $ signs. For instance, the above can be rewritten as:

```
\renewcommand{\seq}{\ensuremath{x_{0},\ldots\,x_{n}}}
\renewcommand{\seqm}[1]{\ensuremath{#1_{0},\ldots\,#1_{n}}}
\seq,\quad\seqm{z} or $\seq,\quad\seqm{z}$
```

$x_0,\ldots\,x_n,\quad z_0,\ldots\,z_n$ or $x_0,\ldots\,x_n,\quad z_0,\ldots\,z_n$

Defining New Environments

In LaTeX 2.09, environments are defined or redefined with the commands:

```
\newenvironment{name}[narg]{begdef}{enddef}
\renewenvironment{name}[narg]{begdef}{enddef}
```

The number of arguments is in the range 0≤*narg*≤9; and, in the case of no parameters, you can omit [0]. Inside the definition part, *begdef*, these parameters are referenced as #1 to #*narg*. If arguments are present, then they are defined when *entering* the environment by specifying them on the command \begin{myenv} as shown below.

```
\begin{myenv}{arg1}...{argk}
```

When *exiting* an environment with the command \end{myenv} no parameters can be specified. Moreover, the parameters specified with the \begin{myenv} command when entering the environment (see above) are no longer available in the definition part *enddef* where you define the actions which should take place when leaving the *myenv* environment.

As with commands, in LaTeX 2ε you can now also define environments with an optional (first) argument.

```
\newenvironment{myenv}[narg][default]{begdef}{enddef}
```

The default for the optional argument is given between the second pair of square brackets [*default*]. Inside the *begdef* part, which is executed when the environment *myenv* is entered, the optional argument can be accessed with #1, while the mandatory arguments (when present) are addressed as #2 to #*narg*. When the *myenv* environment is used without an optional parameter, #1 will contain the string specified as [*default*].

As an example, a variant, deflist, of a description environment will be constructed. The deflist environment behaves somewhat like a standard LaTeX description environment if it is used without an optional argument. If an optional argument is specified, then the width of the description label will be put equal to the width of the argument. Thus, by specifying the widest entry in the list as an optional argument, you ensure that the description parts of all entries line up nicely.

The result below first shows the (default) behaviour of the deflist list and then what it looks like when using the optional argument.

```
\newenvironment{deflist}[1][\quad]%
 {\begin{list}{}{
  \renewcommand{\makelabel}[1]{\textbf{##1}\hfil}%
  \settowidth{\labelwidth}{\textbf{#1}}%
  \setlength{\leftmargin}{\labelwidth+\labelsep}}}
 {\end{list}}
\begin{deflist}
\item[First] This is a short term.
\item[Long term] This is a long term.
\item[Even longer term] A very long term.
\end{deflist}
\begin{deflist}[Even longer term]
    .....
\end{deflist}
```

First This is a short term.

Long term This is a long term.

Even longer term A very long term.

First This is a short term.

Long term This is a long term.

Even longer term A very long term.

2.1.5 Playing with lengths

Lengths can be defined, set and changed by the following commands.

`\newlength{`*cmd*`}`	`\setlength{`*cmd*`}{`*len*`}`
`\addtolength{`*cmd*`}{`*len*`}`	
`\settowidth{`*cmd*`}{`*text*`}`	`\width`
`\settoheight{`*cmd*`}{`*text*`}`	`\height`
`\settodepth{`*cmd*`}{`*text*`}`	`\depth`
	`\totalheight`

The new `\settoheight` and `\settodepth` commands, in analogy with the `\settowidth` command, already present in LaTeX 2.09, allow one to "measure" the height and depth of some TeX material. The lengths `\width`, `\height`, `\depth`, and `\totalheight` are also new in LaTeX 2_ε, and can be used inside the box commands described in the next section.

For ease of reference an overview of TeX's units of length is given below.

`sp` scaled point (65536 sp = 1 pt) TeX's smallest unit.
`pt` point = $\frac{1}{72.27}$ in = 0.351 mm
`bp` big point (72 bp = 1 in), also PostScript point
`dd` Didôt point $= \frac{1}{72}$ of a French inch, $=0.376$ mm
`mm` millimeter = 2.845 pt
`pc` pica = 12 pt = 4.218 mm
`cc` cicero = 12 dd = 4.531 mm
`cm` centimeter = 10 mm =2.371 pc
`in` inch = 25.4 mm = 72.27 pt = 6.022 pc
`ex` height of a small "x" for the current font
`em` width of capital "M" in current font
`mu` math mode unit (18 mu = 1 em)

The following lines show how length commands are created, defined, changed, and used. They work, most of the time, both for rigid and rubber lengths.

```
\newlength{\Mylen} Mylen = \the\Mylen
\setlength{\Mylen}{10mm} Mylen = \the\Mylen
\setlength{\Mylen}{5mm plus 1mm minus .5mm}
\par Mylen = \the\Mylen  % Use a rubber length
\setlength{\Mylen}{1em} One em is \the\Mylen;
\addtolength{\Mylen}{1pc} add one pica \the\Mylen.
\settowidth{\Mylen}{May} The width is \the\Mylen
\settowidth{\Mylen}{\Large May} and now \the\Mylen.

\settoheight{\Mylen}{May} The height is \the\Mylen
\settoheight{\Mylen}{\Large May} and now \the\Mylen.

\settodepth{\Mylen}{May} The depth is \the\Mylen
\settodepth{\Mylen}{\Large May} and now \the\Mylen.
```

Mylen = 0.0pt
Mylen = 28.45274pt
Mylen = 14.22636pt plus 2.84526pt minus 1.42262pt

One em is 10.0pt; add one pica 22.0pt.

The width is 18.32983pt and now 26.39496pt.
The height is 6.68pt and now 9.61917pt.
The depth is 2.16492pt and now 3.11748pt.

"Rubber" (variable) lengths are very useful for placing information on the page.

`\fill`

This is a rubber length with a natural length of zero. It can stretch to any positive value and its value should not be changed!

`\stretch{`*dec num*`}`

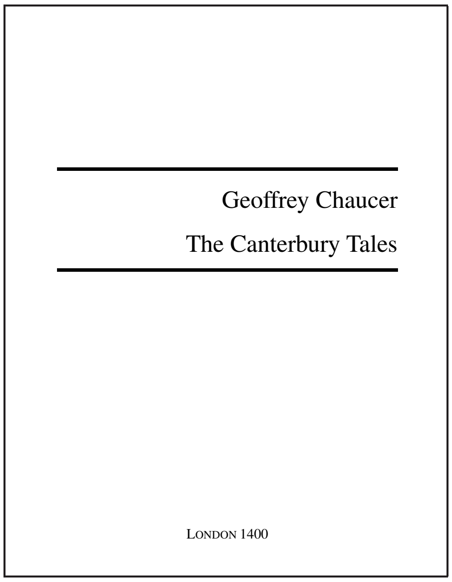
This is a more useful rubber length, since `\fill` is equivalent to `\stretch{1}`. More generally, `\stretch{`*dec num*`}` has a stretchability of *dec num* times `\fill`. It can be used to fine-tune the positioning of text horizontally or vertically.

Examples of the use of these stretchable lengths for controlling the horizontal and vertical page layout are given below.

```
\newcommand{\HS}[1][1.]{\hspace{\stretch{#1}}}
\begin{center}
left \hfill                          right\\
left \HS[.5]\fbox{$\frac{1}{3}$}\hfill right\\
left \HS         middle \hfill         right\\
left \hrulefill\ middle \hrulefill\    right\\
left \dotfill\                         right\\
left \dotfill\ \HS[.5] \dotfill\       right\\
left \dotfill\ \HS     \dotfill\       right\\
left \dotfill\ \HS[2.] \dotfill\       right
\end{center}
```

left right
left $\boxed{\frac{1}{3}}$ right
left middle right
left ______ middle ______ right
left .... right
left .... right
left .... right
left .... right

```
\documentclass{article}
\usepackage{times}
\thispagestyle{empty}
\newcommand{\HRule}{\rule{\linewidth}{1mm}}
\setlength{\parindent}{0mm}
\setlength{\parskip}{0mm}
\begin{document}
  \vspace*{\stretch{1}}
  \HRule
  \begin{flushright}
    \Huge Geoffrey Chaucer\\[5mm]
          The Canterbury Tales
  \end{flushright}
  \HRule
  \vspace*{\stretch{2}}
  \begin{center}
    \Large\textsc{London 1400}
  \end{center}
\end{document}
```



Page Markup—Several Kinds of Boxes

Boxes are at the very heart of TeX's basic typesetting paradigm, and LaTeX provides several commands which make it easy to make use of this functionality.

`\mbox{`*text*`}`	`\makebox[`*width*`][`*pos*`]{`*text*`}`
`\fbox{`*text*`}`	`\framebox[`*width*`][`*pos*`]{`*text*`}`

In addition to centering the text with positional argument `[c]` (the default), you can position the text flush left (`[l]`) or flush right (`[r]`). LaTeX 2ε also offers you an `[s]` specifier that will stretch your *text* from the left margin to the right margin of the box provided it contains some stretchable space. As already mentioned in the previous section, LaTeX 2_ε also allows you to make use of four special length parameters inside the *width* argument of the box commands: `\width`, `\height`, `\depth`, and `\totalheight`. These parameters specify the natural size of the *text*, where `\totalheight` is the sum of `\height` and `\depth`.

The examples below show how these various parameters are used to control the layout in the box. Note that use is also made of the calc package, which allows arithmetic operations in the arguments of the commands.

```
\framebox{A few words of advice}           \par
\framebox[\width + 6mm][s]{A few words of advice}
\par  \framebox[1.5\width]{A few words of advice}
```

A few words of advice
A few words of advice
A few words of advice

`\rule[`*lift*`]{`*width*`}{`*total˙height*`}`

Rules come in handy for controlling the height of a box. Together with the new LaTeX commands for measuring the height and depth of your boxes, they allow you to perform micro-typographic adjustments for tuning the visual presentation of your document elements.

```
\newsavebox{\Maybox}\savebox{\Maybox}{\Large May}
\newlength{\Mdp}\settodepth{\Mdp}{\usebox{\Maybox}}
\newlength{\Mht}\settoheight{\Mht}{\usebox{\Maybox}}
\addtolength{\Mht}{\Mdp}

\framebox[2.6\width+1em][s]{\usebox{\Maybox}}\quad
\framebox[2.6\width+1em][s]{\usebox{\Maybox}\rule[-2\Mdp]{0mm}{2\Mht}}
```

May   May

Zero-width boxes are also useful in other circumstances.

```
\begin{center}
A centred sentence.\makebox[0cm][l]{$^{123}$}\\
Some more text in the middle.               \\
\makebox[0cm][r]{$^{321}$}A centred sentence.\\
\end{center}
\noindent\makebox[0cm][r]{\(\Leftrightarrow\)}%
As seen in the margin of the current line, boxes
with a vanishing width can stick out in the margin.
```

A centred sentence.^{123}
Some more text in the middle.
^{321}A centred sentence.

$\Leftrightarrow$As seen in the margin of the current line, boxes with a vanishing width can stick out in the margin.

Moving boxes

Boxes can be moved up or down by the command:

`\raisebox{`*lift*`}[`*depth*`][`*height*`]{`*contents*`}`

The simple example below shows its principle of use.

```
\begin{flushleft}
x111x \raisebox{-1ex}{downward} x222x     \\
x333x \raisebox{1ex}{upward} x444x        \\[1em]
x111x \raisebox{-1ex}[0cm][0cm]{downward} x222x\\
x333x \raisebox{1ex}[0cm]{upward} x444x
\end{flushleft}
```

x111x downward x222x
x333x upward x444x

x111x downward x222x
x333x upward x444x

A more useful example is the generation of text "between" two rows in a table (by "hiding" the boxes' content from TeX.)

```
\begin{center}
\begin{tabular}{|c|c|c|}              \hline
        & \multicolumn{2}{c|}{title}\\\cline{2-3}
\raisebox{1.5ex}[0cm][0cm]{100}
        & A          & B              \\\hline
20000000 & 10        & 10             \\\hline
\end{tabular}
\end{center}
```

100	title	
	A	B
20000000	10	10

Finally, when your printer driver allows it, you can rotate boxes. In this case the use of the various box dimension parameters becomes apparent.

```
\newcommand{\DoT}[1]{\rotatebox{45}{#1}}
x1 \DoT{\fbox{Not so Good.}} x2
\DoT{\raisebox{\depth}{\fbox{I like it.}}} x3
\DoT{\raisebox{-\height}{\fbox{Bad Choice.}}} x4
```

x1 Not so Good. x2 I like it. x3 Bad Choice.
x4

Placing parboxes and minipages

In LaTeX 2.09, boxes, which can contain more than one paragraph are defined as follows.

```
\parbox[pos]{width}{text}
\begin{minipage}[pos]{width}
    text
\end{minipage}
```

A simple example of use is the following

```
\parbox{.3\linewidth}{This is the
 contents of the left-most parbox.}
\hfill Centerline \hfill
\parbox{.3\linewidth}{This is the
 right-most parbox. Note that the
 typeset text looks sloppy because
 \LaTeX{} cannot nicely balance the
 material in these narrow columns.}
```

This is the contents of the left-most parbox. Centerline This is the right-most parbox. Note that the typeset text looks sloppy because LaTeX cannot nicely balance the material in these narrow columns.

Generalized parboxes and minipages

Sometimes it is helpful to predefine the vertical dimension of a paragraph box. For this LaTeX 2ε has additional optional arguments for `minipage` and `\parbox`.

```
\parbox[pos][height][inner-pos]{width}{text}
\begin{minipage}[pos][height][inner-pos]{width}
    text
\end{minipage}
```

The *inner-pos* determines the position of *text* within the box. It can be `t`, `c`, `b`, or `s`. If not specified, the value of *pos* will be used. You can think of *height* and *inner-pos* as the vertical equivalent of the *width* and *pos* arguments of a `\makebox`. If you use the `s` position the *text* will be vertically stretched to fill the given *height*. Thus, in this case you are responsible for providing vertically stretchable space if necessary using, for example, `\vspace` or `\vfill` commands.

As with the other box commands you can use `\height`, `\totalheight`, and so on to refer to the natural dimensions of the box when specifying the optional argument.

```
xx \fbox{\parbox[b][1.5\height][s]
      {30mm}{Some text on top. \par\vfill
                In the middle. \par\vfill
                And a few lines on the
                bottom of the box.}}
\fbox{\parbox[b][\height+\baselineskip][s]
      {30mm}{This time a few lines on the
             top of the box. But only one
             line \par\vfill down here.}} xx
```

xx

Some text on top.

In the middle.

And a few lines on the bottom of the box.

This time a few lines on the top of the box. But only one line

down here.

xx

Manipulating Boxed Material

Material can be typeset once and then stored inside a named box, so that its contents can be retrieved later. LaTeX offers the following commands for dealing with this situation.

`\newsavebox{`*cmd*`}`	declare box
`\sbox{`*cmd*`}{`*text*`}`	fill box
`\savebox{`*cmd*`}[`*width*`][`*pos*`]{`*text*`}`	fill box
`\usebox{`*cmd*`}`	use contents
`\begin{lrbox}{`*cmd*`}` *text* `\end{lrbox}`	fill box

Note that the environment `lrbox` is an addition in LaTeX 2_ε. *cmd* should be a box register previously allocated with `\newsavebox`. The environment `lrbox` will save *text* in this box for later use with `\usebox`. Leading and trailing spaces are ignored. Thus, `lrbox` is basically the environment form of `\sbox`. You can make good use of this environment if you want to save the body of some environment in a box for further processing. For example, the following code defines the environment `fminipage` that works like a `minipage` but surrounds its body with a frame. Note the use of the optional argument for controlling the width of the boxed minipage, and the fact that verbatim material can be used inside. To be able to do the arithmetic operations you will also need to have the calc package loaded.

```
\newsavebox{\fminibox}
\newlength{\fminilength}
\newenvironment{fminipage}
    [1][\linewidth]% default width is \linewidth
  {\setlength{\fminilength}%
        {#1-2\fboxsep-2\fboxrule}%
   \begin{lrbox}{\fminibox}%
   \begin{minipage}{\fminilength}}
  {\end{minipage}\end{lrbox}%
   \noindent\fbox{\usebox{\fminibox}}}

\begin{fminipage}
  In this environment verbatim text like \verb=\fminibox= can be used.
\end{fminipage}
```

In this environment verbatim text like `\fminibox` can be used.

```
\begin{fminipage}[.5\linewidth]
  In this environment verbatim text like \verb=\fminibox= can be used.
\end{fminipage}
```

In this environment verbatim text like `\fminibox` can be used.

For hackers only: several optional arguments

What if we want to define a command or environment with, e.g., *two* optional arguments? Suppose we want a list where we are able to specify not only the width of the label, but also whether the list should be "dense" or not, i.e., we want a syntax like:

```
\begin{Description}[<margin>][<style>]
```

In this case we use a trick (and the packages calc and ifthen) and introduce a multiple-step definition. The example also shows how one can parameterize the various typographic parameters for the list so that they can be controlled more easily (e.g., the alignment of the label, its font, and the width of the margin).

```
\newcommand{\Descriptionlabel}[1]{%
       \mbox{\Descriptionfont #1}\hfil}

\newcommand\Descriptionfont{\itshape}
\newcommand\Descriptionmargin{}

\newenvironment{Description}[1][\kern\leftmargin]
 {\renewcommand\Descriptionmargin{#1}\xdescription}
 {\endlist}

\newcommand{\xdescription}[1][normal]{%
 \list{}{\settowidth{\labelwidth}%
  {\mbox{\Descriptionfont\Descriptionmargin}}%
  \setlength{\itemindent}{0pt}%
  \setlength{\leftmargin}{\labelwidth+\labelsep}%
  \let\makelabel\Descriptionlabel
  \ifthenelse{\equal{#1}{compact}}%
   {\setlength{\itemsep}{0pt}%
    \setlength{\topsep}{.5\topsep}}{}%
 }}

Text before text before text before text before
\begin{Description}
  \item[First] This is the first item in the list.
  \item[Veryyy long] This is a veryyy long item.
  \item[] This is an empty item.
\end{Description}

\begin{Description}[Veryyy long]
  ...
```

```
\end{Description}

\begin{Description}[Veryyy long][compact]
  ...
\end{Description}
```

Text before text before text before text before

First This is the first item in the list.

Veryyy long This is a veryyy long item.

This is an empty item.

First This is the first item in the list.

Veryyy long This is a veryyy long item.

This is an empty item.

First This is the first item in the list.
Veryyy long This is a veryyy long item.
This is an empty item.

2.1.6 Font commands—an overview

This section covers the user commands in LaTeX 2_ε for specifying fonts, both in text as in mathematics. We also mention some of the more popular fonts packages and say a few words on compatibility with LaTeX 2.09.

The first question you can naturally ask yourself is why new font commands were introduced at all. To answer this question let us mention that LaTeX 2.09 font commands had a few idiosyncrasies:

- their syntax, i.e., `{\it foo}` rather than `\it{foo}`, which is unlike the syntax of (most) other LaTeX commands (safe the size-changing series), which are specified with arguments;
- the font commands were not *orthogonal*, e.g., `\bf\sf` produces medium-weight sans, i.e., only the inner font command is honored;
- some fonts substitutions were taking place "behind our backs", e.g., `\tiny\tt` produces tiny roman, since it was assumed that at such a small size the difference is hardly visible, so that one can as well use a font already loaded;
- italic corrections must be introduced by hand, e.g., one has to write `{\em my text\/}`, and even this is not correct in all circumstances.

LaTeX 2_ε addresses these problems by introducing the following new text font commands:

`\textmd`{This is medium text}

`\textbf`{**This is bold text**}

`\textup`{This is upright text}

`\textit`{*This is italic text*}

`\textsl`{*This is slanted text*}

`\textsc{`THIS IS SMALL CAPS TEXT`}`

`\textrm{`This is roman text`}`

`\textsf{`This is sans text`}`

`\texttt{This is typewriter text}`

Plus `\emph{`*This is emphasized text*`}`.
The size changing commands remain unchanged (i.e., `\large`, `\scriptsize`, etc. are still valid).
These commands do not have the problems of the LaTeX 2.09 commands, because:

- their syntax is the same as for the other LaTeX commands;
- `\textbf{\textsf{`**text**`}}` produces bold sans;
- `{\tiny\texttt{text}}` produces tiny typewriter;
- `\emph{`*text*`}` does *not* need `\/`.

Note that there are still some restrictions, for instance, `\textbf{\texttt{text}}` produces medium typewriter for lack of a bold Computer Modern typewriter font, but at least LaTeX 2ε warns you about the substitution.
In the area of math fonts, LaTeX 2ε provides the following new commands:

`\mathnormal{`$This is normal math italic$`}`

`\mathcal{`$\mathcal{MATH\ CALLIGRAPHIC}$`}`

`\mathrm{`$\mathrm{This\ is\ roman\ in\ math}$`}`

`\mathbf{`$\mathbf{This\ is\ bold\ in\ math}$`}`

`\mathsf{`$\mathsf{This\ is\ sans\ in\ math}$`}`

`\mathit{`$\mathit{This\ is\ text\ italic\ in\ math}$`}`

`\mathtt{`$\mathtt{This\ is\ typewriter\ in\ math}$`}`

Note that these commands do not work outside mathematics.
It is now relatively easy (if you have the fonts) to replace Computer Modern with other font families. Various packages for popular fonts are already available, for example:

- In the area of PostScript fonts `\usepackage{times}` provides Adobe Times, `\usepackage{palatino}` Adobe Palatino, `\usepackage{lucidbrb}` Y&Y's LucidBright and LucidaNewMath, etc.;
- `\usepackage{amssymb}` provides the AMS fonts;
- `\usepackage{pandora}` allows you to use the Pandora fonts;
- `\usepackage{euler}` lets you experiment with Hermann Zapf's Euler font family.

LaTeX 2.09's old font commands (`\rm`, `\bf`, etc.) are still available in LaTeX 2ε, but they are *not* part of the "kernel". They are now defined in the document class files, where the definitions of the size changing commands, like `\huge`, `\tiny`, have always resided. It is thus up to the document designer to define how the old font commands behave. Note, however, that for the "standard classes" (article, book, etc.) the old font commands behave as they did in LaTeX 2.09.
One more word about about LaTeX 2.09 compatibility. A document beginning with `\documentstyle` is run in *compatibility mode*, which emulates LaTeX without NFSS. If you want to emulate LaTeX with NFSS you should say:

```
\documentstyle[newlfont]{...}
```

2.1.7 Standard Classes in LaTeX 2ε

This section discusses the files that come with the LaTeX 2ε distribution and lists some of the packages which are already adapted to LaTeX 2ε.

Files associated to LaTeX 2ε are characterized by the extensions:

name`.cls` for class files;
name`.clo` for external option files;
name`.sty` for package files
name`.cfg` for runtime configuration files

The "standard" document classes distributed with LaTeX 2ε are article, report, book, letter, slide, proc, and ltxdoc. Below, we say a few words about each one of them.

article, report, book

- they behave like the old LaTeX 2.09 styles;
- twocolumn and openbib are now internal options;
- a set of new internal options was added: a4paper, a5paper, b5paper, letterpaper, legalpaper, executivepaper, landscape.

letter

- it behaves like the old style;
- it has a set of new internal options: a4paper, a5paper, b5paper, letterpaper, legalpaper, executivepaper.

slide

- it behaves like the old style, but used with LaTeX 2ε;
- it supports local font configuration by looking for `sfonts.cfg`;
- it has a set of new internal options: a4paper, a5paper, b5paper, letterpaper, legalpaper, executivepaper, landscape;
- the option twocolumn is not supported.

proc

- it is no longer an option but a *document class*;
- it is built on the article class;
- it disallows options a5paper, b5paper, onecolumn, titlepage.

ltxdoc

- it is used to format the LaTeX 2ε source code;
- it is built on article and requires the doc package;
- it looks for the configuration file `ltxdoc.cfg`;
- it defines the commands `\DocInclude` and `\GetFileInfo`;
- it disallows the option a5paper.

2.1.8 Standard LaTeX 2ε packages

The following packages are distributed with LaTeX:

alltt Defines the `alltt` environment, which is like the `\verbatim` environment except that `\`, `{`, and `}` have their usual meanings (see [1]).

doc The basic package for typesetting the documentation of LaTeX programs (see [2]). *The LaTeX Companion*.

exscale Provides scaled versions of the math extension font (see [2]).

fontenc Used to specify which font encoding LaTeX should use. It is described in `ltoutenc.dtx`.

graphpap Defines the `\graphpaper` command to be used in a `picture` environment.

ifthen Defines some Boolean constructs (see [2]).

inputenc Allows one to specify which input encoding LaTeX should use. It is described in `inputenc.dtx`.

latexsym LaTeX 2ε no longer loads the LaTeX symbol font by default. To access it, you should use the `latexsym` package (see [2]).

makeidx Activates the commands for producing indexes. (see [1,2] and section 4.3.1).

newlfont Emulates font commands of LaTeX 2.09 with the New Font Selection Scheme, Version 1 (see [2]).

oldlfont Emulates font commands of standard LaTeX 2.09 (see [2]).

showidx Causes the argument of each `\index` command to be printed on the page where it occurs (see [1,2]).

syntonly Processes a document without typesetting it (see [2]).

tracefnt Allows one to control how much information about LaTeX's font loading is displayed (see [2]).

2.1.9 LaTeX 2ε packages supported by the LaTeX3 team

A series of add-on packages supported by the LaTeX3 team is available. Especially the graphics and tools collections are interesting complements. Together with the amstex and babel packages they are mentioned as "standard packages" in section C.5.2 of [1].

amslatex Advanced mathematical typesetting from the American Mathematical Society. This is includes the amsmath package; it provides many commands for typesetting mathematical formulas of higher complexity (see [2] or the documentation coming with the software).

babel Support for typesetting in many languages (see [2]).

graphics This bundle includes the graphics package allowing inclusion and transformation of graphics (see section 2.2) and the color package providing support for typesetting in colour (see section 2.3, both these packages are also described in [1]).

mfnfss Contains the set-up (except the fonts themselves) for typesetting with a large range of bit-map (Metafont) fonts.

psnfss Contains the set-up (except the fonts themselves) for typesetting with a large range of Type 1 (PostScript) fonts.

tools A series of packages written by the LaTeX3 project team (see below).

Tools

This collection of packages written and supported by the LaTeX3 project team includes those listed below:

array Extended versions of the environments `array` and `array` environments, with many extra features (see [2]).

dcolumn Alignment on "decimal points" in tabular entries (requires the array package; see [2]).

delarray Adds "large delimiters" around arrays (requires the array package; see [2]).

hhline Finer control over horizontal rules in tables (requires the array package; see [2]).

longtable Multi-page tables (does not require array, but uses its extended features if it is loaded; see [2]).

tabularx Defines a `tabularx` environment that is similar to `tabular*`, but it modifies the column widths, rather than the inter-column space, to achieve the desired table width (see [2]).

afterpage Place text after the current page (see [2]).

enumerate Extended version of the `enumerate` environment (see [2]).

fontsmpl Package and test file for producing "font samples".

ftnright Place all footnotes in the right-hand column in two-column mode (see [2]).

indentfirst Indent The first paragraph of sections, etc. (see [2]).

layout Show the page layout defined by the current document class (see [2]).

multicol Typeset text in columns, with the length of the columns "balanced" (see [2]).

rawfonts Preload fonts using the old internal font names of LaTeX 2.09.

somedefs Selective handling of package options (used by the rawfonts package).

showkeys Prints the "keys" used by cross-reference commands `\label`, `\ref`, `\cite`, etc.; useful whilst drafting.

theorem Flexible declaration of "theorem-like" environments (see [2]).

varioref "Smart" handling of page references (see [2]).

verbatim Flexible extension of the verbatim environment (see [2]).

xr Cross reference other "external" documents (see [2]).

xspace "Smart space" command that helps you to avoid the common mistake of missing spaces after command names (see [2]).

2.1.10 Miscellaneous goodies

This section describes some features which are perhaps not used every day, but which can come in handy for solving certain practical document preparation problems.

Controlling page breaks

Sometimes, when preparing the *final* version of your document, you might need to help LaTeX break the pages in a suitable way. LaTeX 2.09 had commands like `\clearpage`, `\samepage`, etc., while LaTeX 2ε provides, in addition, commands which increase or decrease the height of the *current* page from its "natural" height `\textheight` by an amount *size*.

```
\enlargethispage{size}
\enlargethispage*{size}
```

For example, `\enlargethispage{-\baselineskip}}` decreases the length of the current page by one line, while `\enlargethispage*{2\baselineskip}}` makes it two lines longer than usual.
The starred form also shrinks any vertical white space on the page as much as possible, so as to fit the maximum amount of text onto the page.

Floats

A new command and a new "float specifier" will allow you more control over LaTeX's float placement algorithm.

```
\suppressfloats[placement]
```

This command stops any further floating environments from being placed on the current page. The optional argument *placement* can be either `t` or `b` (not both), and in this case the restriction applies only to putting further floats at the top or at the bottom.

Extra float placement specifier: `!`

This can be used, along with at least one of `h`, `t`, `b` and `p`, in the float placement optional argument.
If a `!` is present then, just for this particular float, whenever it is processed by the float mechanism the following are ignored:

- all restrictions on the number of floats which can appear;
- all explicit restrictions on the amount of space on a text page which may be occupied by floats or must be occupied by text.

The mechanism will, however, still attempt to ensure that pages are not overfull and that floats of the same type are printed in the correct order.
Note that its presence has no effect on the production of float pages.
A `!` placement specifier overrides the effect of any `\suppressfloats` command for this particular float.

2.2 Graphics commands in LaTeX 2ε

2.2.1 Introduction

LaTeX 2ε provides a generalized driver-independent interface for the inclusion of external graphic material, as well as for scaling and rotation of LaTeX boxes. These features are not in the LaTeX 2ε kernel, so you have to load the standard supported graphics or graphicx extension packages. The color extension package, that will be discussed in detail in one of the next issues of the CNL, implements basic colour functions. These packages rely on features that are not in TeX itself, but which must be supplied by the "driver" used to print the dvi file[2]. Inside a `picture` environment LaTeX 2ε offers the possibility to draw Bezier curves (with the `\qbezier` command) or grids (by loading the graphpap package).

[2]Unfortunately not all drivers support the same features, and even the internal method of accessing these extensions varies between drivers. Consequently all these packages take options such as "`dvips`" to specify which driver is being used.

2.2.2 The graphics package

The graphics package (and its extension graphicx) allow the user to scale, rotate, and reflect LaTeX material, or include graphics prepared with other programs.

LaTeX will leave the necessary space for the material after applying the geometric transformation, but it is the device driver (e.g. dvips, xdvi, etc.) which must be capable of performing the transformation in question. Some drivers, such as previewers, might therefore display material overlapping the surrounding text.

Scaling a LaTeX box

The `\scalebox` command lets you magnify or reduce text or other LaTeX material by a scale factor.

`\scalebox{`*scalefact*`}{`*contents*`}`

When specified with two (mandatory) arguments, the first one (*scalefact*) specifies the factor by which both dimensions of the material (*contents*) have to be scaled.

As an example consider the following:

```
\scalebox{2}{Text scaled by a factor or two.}\\
             Text scaled by a factor of one.\\
\scalebox{0.5}{Text scaled by a factor of one half.}
```

Text scaled by a factor or two.

Text scaled by a factor of one.

Text scaled by a factor of one half.

If a supplementary optional argument is present, then this is used to specify a separate vertical scaling factor.

`\scalebox{`*h-scale*`}[`*v-scale*`]{`*contents*`}`

This is demonstrated in the following examples, which also show how multiple lines can be scaled by using the standard LaTeX `\parbox` command.

```
\framebox{\scalebox{2}{\parbox{.5in}{America \&\\Europe}}}
\framebox{\scalebox{2}[.5]{\parbox{.5in}{America \&\\Europe}}}
```

America & Europe   America & Europe

`\reflectbox{`*contents*`}`

This is an abreviation for `\scalebox{-1}[1]{contents}`.

```
America?\reflectbox{America?}\quad
America?\scalebox{-1}[1]{America}\quad
America?\scalebox{-1}{America}\quad
America?\scalebox{-1}[-1]{America}
```

America??aciremA   America?aciremA   America?aciremA   America?aciremA

Resizing to a given size

It is also possible to specify that LaTeX material should be typeset to a fixed horizontal or vertical dimension.

`\resizebox*{`*h-dim*`}{`*v-dim*`}{`*LaTeX box*`}`

When the aspect ratio of the material should be maintained, then it is enough to specify one of the dimensions, replacing the other one with a "`!`" sign.

```
\framebox{\resizebox{5mm}{!}{\parbox{14mm}{London, \\Berlin \&\\Paris}}}
\framebox{\resizebox{!}{10mm}{\parbox{14mm}{London, \\Berlin \&\\Paris}}}
```

London,
Berlin &
Paris

London,
Berlin &
Paris

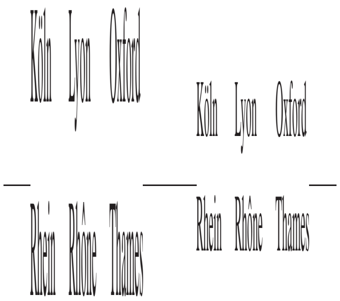
When explicit dimensions for both *h-dim* and *v-dim* are supplied, then one can distort the contents. In the following the baseline will be indicated by and horizontal rule drawn with the `\HR` command.

```
\newcommand{\HR}{\rule{1em}{0.4pt}}
\HR\begin{tabular}{@{}lll@{}}
  K\"oln & Lyon    & Oxford \\
 Rhein   & Rh\^one & Thames
 \end{tabular}\HR
\HR\resizebox{2cm}{.5cm}{\begin{tabular}{@{}lll@{}}
  K\"oln & Lyon    & Oxford \\
 Rhein   & Rh\^one & Thames
 \end{tabular}}\HR
```

Köln	Lyon	Oxford	Köln	Lyon	Oxford
Rhein	Rhône	Thames	Rhein	Rhône	Thames

The unstarred form `\resizebox` bases its calculations on the height of the LaTeX material, while the starred `\resizebox*` command takes into account the total height (the depth plus the height) of the LaTeX box. The next `\parbox` examples, which have a large depth, show the difference.

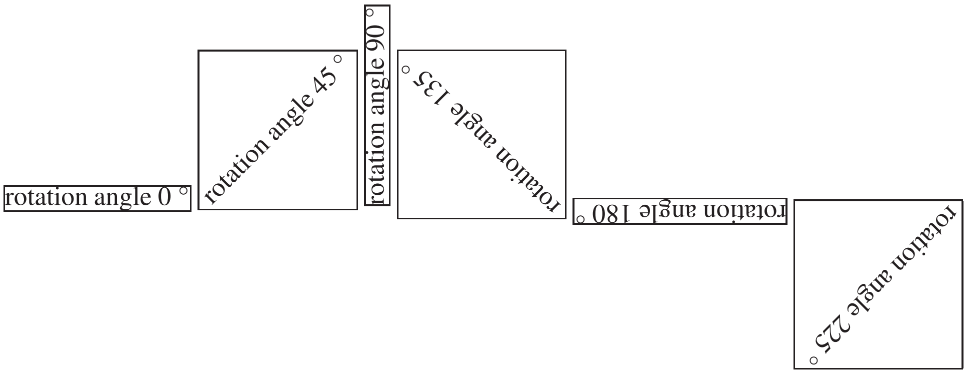
```
\HR\resizebox{15mm}{30mm}{\begin{tabular}{@{}lll@{}}
                            K\"oln & Lyon    & Oxford \\
                            Rhein   & Rh\^one & Thames
                          \end{tabular}}\HR
\HR\resizebox*{15mm}{30mm}{\begin{tabular}{@{}lll@{}}
                            K\"oln & Lyon    & Oxford \\
                            Rhein   & Rh\^one & Thames
                           \end{tabular}}\HR
```



Rotating a LaTeX box

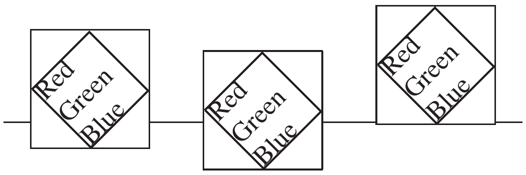
`\rotatebox{`*angle*`}{`*contents*`}`

LaTeX material can be rotated over an angle with the `\rotatebox` command, whose second argument (*contents*) is typeset inside a LaTeX box and rotated through *angle* degrees counter-clockwise around the reference point.



It is important to realize where the reference point is; the following examples show it can be shifted, using standard `\parbox` and `\makebox` parameters. Notice the effect of the placement parameter on the `\parbox` command.

```
\HR\framebox{\rotatebox{45}{\framebox{\parbox{3em}{Red\\Green\\Blue}}}}\HR
\HR\framebox{\rotatebox{45}{\framebox{\parbox[t]{3em}{Red\\Green\\Blue}}}}\HR
\HR\framebox{\rotatebox{45}{\framebox{\parbox[b]{3em}{Red\\Green\\Blue}}}}\HR
```

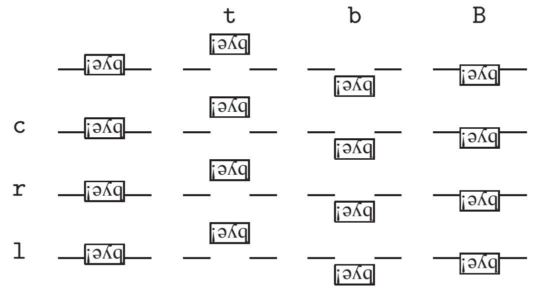


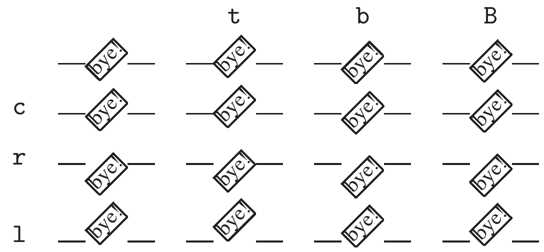
With the extended graphics package graphicx one has more flexibitity to specify the point around which the rotation is to take place by using *key val* pairs.

`\rotatebox[`*key val list*`]{`*angle*`}{`*LaTeX box*`}`

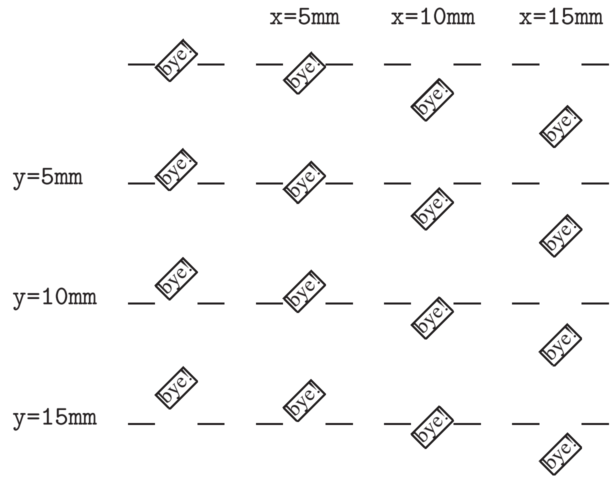
One of the possible keys is `origin` and for the values you can choose one or two amongst the following list:

horizontal	`l`	left	`r`	right	`c`	centre
vertical	`t`	top	`b`	botom	`B`	baseline





One can also supply x and y coordinates with `x=`*dim*`, y=`*dim*, which specify the new point (relative to the reference point) around which the rotation is to take place.



The interpretation of the *angle* argument of `\rotatebox` can be controlled by the *units* keyword, which specifies a factor to be applied to the angle, e.g., `units= 6.283185` changes the degree specification to radians. Similarly, the direction of rotation can be changed to clockwise by putting `units=-360` in `\rotatebox`'s optional argument.

Rotating tabular material

Tabular material can also be rotated. The examples below show how the distance between the columns and the vertical placement of the table can be controlled by the use of zero width or zero height rules.

```
\newcommand{\IR}[1]{\rotatebox{45}{\ #1}}
\begin{tabular}{rrr}
\IR{Column 1} & \IR{Column 2} & \IR{Column 3} \\\hline
1& 2& 3 \\ 4& 5& 6 \\ 7& 8& 9 \\\hline
\end{tabular}\qquad
```

```
% Zero height rule of width of one quad
\renewcommand{\IR}[1]{\rule{1em}{0pt}\makebox[0cm][c]{\rotatebox{45}{\ #1}}}
\begin{tabular}{rrr}
\IR{Column 1} & \IR{Column 2} & \IR{Column 3} \\\hline
1& 2& 3 \\ 4& 5& 6 \\ 7& 8& 9 \\\hline
\end{tabular}
```

Column 1	Column 2	Column 3
1	2	3
4	5	6
7	8	9

Column 1	Column 2	Column 3
1	2	3
4	5	6
7	8	9

Rotation inside tables can be nested, as shown below.

```
\setlength{\extrarowheight}{4pt}
\HR\rotatebox{90}{%
  \begin{tabular}{|l@{\quad}r|}
    \hline
    \em Word    &
      \raisebox{\depth}{%
      \rotatebox{90}{Occurrences}} \\[1mm]
    \hline
    hello & 33\\ goodbye & 34\\ \hline
  \end{tabular}}\HR
```

Word	Occurrences
hello	33
goodbye	34

A more complex example is given in table 2.1 on the facing page. The whole table is rotated by a `\rotatebox` command. Note the use of another `rotatebox` command to generate vertically running text. By specifying a zero depth and height for the result of the `\raisebox` command LaTeX is made to think that 'Format classes' takes up no vertical space.

Rotation inside tables

As shown above, material inside a `tabular` or `array` environment can be rotated at will. This functionality can be used, e.g., to generate (rotated) braces, that span multiple columns or rows.

The building block in the following examples and in Table 2.2 is the `\Bpara` macro whose definition is shown below. The principle is to make a `}` in math mode, but hide it from TeX. That is why we use a `picture` environment of vanishing height and width. It contains a zero-width `\makebox` (to hide the horizontal dimension), that contains the rotated material, whose height is hidden from LaTeX by the `\raisebox` command specifying a zero width and depth. The (zero-width) `\rule` controls the height of the `}` in math mode (the `\left.` command is needed to balance the parentheses). Similar commands can be constructed for other large symbols.

```
\setlength{\unitlength}{1mm}
% #1 x coordinate of reference point of brace
% #2 y coordinate of reference point of brace
% #3 rotation angle for large brace
% #4 total height of brace
\newcommand{\Bpara}[4]{%
  \begin{picture}(0,0)%
   \put(#1,#2){\rotatebox{#3}{\raisebox{0mm}[0mm][0mm]{%
                   $\left.\makebox[0mm]{\rule{0mm}{#4}}\right\}$}}}
  \end{picture}%
}
```

ISO paper formats (mm)				
		A series	B series	C series
Format classes	0	841×1189	1000×1414	917×1297
	1	594×841	707×1000	648×917
	2	420×594	500×707	458×648
	3	297×420	353×500	324×458
	4	210×297	250×353	229×324
	5	148×210	176×250	162×229
	6	105×148	125×176	114×162
	7	74×105	88×125	81×114
	8	52×74	62×88	57×81

```
\renewcommand{\arraystretch}{1.1}
\setlength{\tabcolsep}{2mm}
\rotatebox{-90}{%
\begin{tabular}{|l|l*3{r@{$\times$}l}|}
                                \hline
\multicolumn{8}{|c|}{ISO paper formats (mm)}
                              \\\hline
&&\multicolumn{2}{c}{A series}
 &\multicolumn{2}{c}{B series}
 &\multicolumn{2}{c|}{C series}\\\cline{2-8}
&0&841&1189&1000&1414&917&1297 \\\cline{2-8}
&1&594&841 & 707&1000&648&917  \\\cline{2-8}
&2&420&594 & 500&707 &458&648  \\\cline{2-8}
&3&297&420 & 353&500 &324&458  \\\cline{2-8}
&4&210&297 & 250&353 &229&324  \\\cline{2-8}
&5&148&210 & 176&250 &162&229  \\\cline{2-8}
&6&105&148 & 125&176 &114&162  \\\cline{2-8}
&7& 74&105 &  88&125 & 81&114  \\\cline{2-8}
\raisebox{8mm}[0mm][0mm]{%
    \rotatebox{90}{%
       \hspace*{8mm}Format classes}}%}%
&8& 52&74  &  62&88  & 57&81   \\\hline
\end{tabular}
```

Table 2.1: Rotating tabular information

```
\begin{tabular}{||l|r@{\,--\,}l|l|l||lll||ll|}
  \hline
  1.4$\cdot10^{-4}$ & 0.02 & 0.06 & 0.03 & 0.64 & 0.935 & 0.066 & 0.020 & &            \\
  2.8$\cdot10^{-4}$ & 0.04 & 0.09 & 0.07 & 0.65 & 0.922 & 0.051 & 0.030 & &            \\
  4.5$\cdot10^{-4}$ & 0.04 & 0.15 & 0.11 & 0.64 & 0.865 & 0.038 & 0.021 &
                                                   \Bpara{-5}{0}{0}{8ex}0.018 & 0.097\\
  6.7$\cdot10^{-4}$ & 0.06 & 0.22 & 0.15 & 0.61 & 0.897 & 0.036 & 0.015 & &            \\
  9.0$\cdot10^{-4}$ & 0.15 & 0.25 & 0.21 & 0.67 & 0.894 & 0.027 & 0.020 & &            \\
  1.5$\cdot10^{-3}$ & 0.15 & 0.3  & 0.34 & 0.63 & 0.888 & 0.012 & 0.021 & 0.006 & 0.048\\
  3.0$\cdot10^{-3}$ & 0.15 & 1.2  & 0.61 & 0.61 & 0.917 & 0.009 & 0.009 & 0.003 & 0.030\\
  \hline
\end{tabular}
```

$1.4{\cdot}10^{-4}$	0.02 – 0.06	0.03	0.64	0.935	0.066	0.020		
$2.8{\cdot}10^{-4}$	0.04 – 0.09	0.07	0.65	0.922	0.051	0.030		
$4.5{\cdot}10^{-4}$	0.04 – 0.15	0.11	0.64	0.865	0.038	0.021	0.018	0.097
$6.7{\cdot}10^{-4}$	0.06 – 0.22	0.15	0.61	0.897	0.036	0.015		
$9.0{\cdot}10^{-4}$	0.15 – 0.25	0.21	0.67	0.894	0.027	0.020		
$1.5{\cdot}10^{-3}$	0.15 – 0.3	0.34	0.63	0.888	0.012	0.021	0.006	0.048
$3.0{\cdot}10^{-3}$	0.15 – 1.2	0.61	0.61	0.917	0.009	0.009	0.003	0.030

Table 2.2: Constructing brackets spanning several rows

```
\[ \begin{array}{ccccc}
     a_{11}                                  \\
     a_{21} & a_{22}                         \\
     a_{31} & a_{32} &
              a_{33}\Bpara{0}{2}{57}{6.7em}    \\
     a_{41} & a_{42} & a_{43} & a_{44}         \\
     a_{51} & a_{52} &
              a_{53} \Bpara{-6}{-1}{-90}{6em}
                               & a_{54} & a_{55}
   \end{array}
\]
```

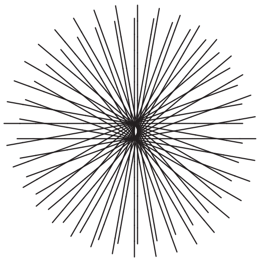
$$\begin{array}{ccccc} a_{11} \\ a_{21} & a_{22} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} & a_{44} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{array}$$

```
\begin{tabular}{|c|c|c|}                 \hline
\multicolumn{2}{|c|}{}          & 13   \\\cline{3-3}
\multicolumn{2}{|c|}{\raisebox{-1.5ex}[0mm][0mm]{%
            \Large A \Bpara{3}{7.5}{180}{2.3ex}}}
                                & 23   \\\cline{3-3}
\multicolumn{2}{|c|}{}          & 33   \\\hline
41           & 42               & 43   \\\hline
\end{tabular}
```

		13
A		23
		33
41	42	43

Using the ifthen package one can show the effect of rotating a double vertical bar over 360 degrees, in 10 degree steps.

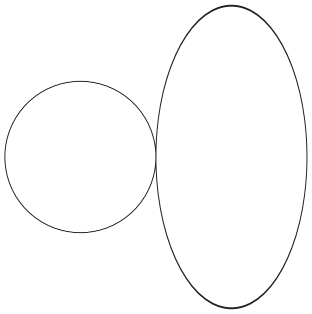
```
\begin{center}
\newcounter{Angle}
\whiledo{\value{Angle}<360}{%
         \Bpara{0}{0}{\theAngle}{5em}%
         \addtocounter{Angle}{10}
        }
\end{center}
```

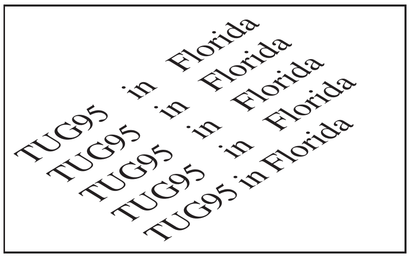


2.2.3 Combining effects

Rotation and scaling commands can be combined and nested *ad lib.*; we can also apply them to LaTeX picture environment objects.

```
\setlength{\unitlength}{1mm}
\begin{picture}(50,40)
 \put(15,20){\circle{20}}
 \put(35,20){\scalebox{1}[2]{\circle{20}}}
\end{picture}\qquad
\framebox{\resizebox{5cm}{25mm}{\rotatebox{45}{\parbox{3cm}{%
   TUG95 in Florida TUG95 in Florida TUG95 in Florida
   TUG95 in Florida TUG95 in Florida}}}
}
```





2.2.4 Including Graphics Files

To make it easier for the user to include any kind of graphics file that can be handled by the different drivers, LaTeX 2ε offers a uniform syntax for the inclusion of such files. Generally, it relies on the file having an extension that identifies the type of file. Driver configuration files define a collection of file extensions that the driver can handle. This list can be extended using the declarations described below. Although most of the examples below use PostScript inclusions, the system also copes with files of type TIFF, PCX, Macdraw, etc. Of course appropriate drivers must also be available for each case.

The basic graphics package

A graphics file *file* can be included using the graphics package with the command:

`\includegraphics*[`*llx,lly*`][`*urx,ury*`]{`*file*`}`

Without optional arguments the size of the graphic will be determined by reading the external file *file* (containing the graphics itself, or a description thereof, see below).

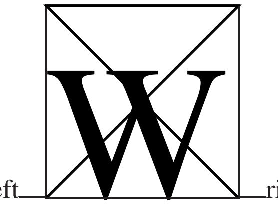
If [*urx,ury*] is present, then it should specify the coordinates of the top right corner of the image, as a pair of TeX dimensions. The default units are big (PostScript) points, i.e., `[1in,1in]` and `[72,72]` are equivalent. If only one optional argument is given, the lower left corner of the image is assumed to be at `[0,0]`. Otherwise, [*llx,lly*] specifies the coordinates of that point.

The starred form of the command "clipped" the graphics image to the size of the size (bounding box) specified. In the normal form (without the `*`) of the `\includegraphics` command, any part of the graphics image that falls outside the specified bounding box will over-print the surrounding text.

To see the effect of the various arguments, we will show the effect of each one of them. In the following the framebox corresponds to the "bounding box", and the baseline is indicated by the horizontal rules. We use a small PostScript program (in a file `w.eps`), that paints a large uppercase letter A, and a few lines. Its source is shown below. Note the `BoundingBox` command, which stipulates that the image starts at the point `100 100` (in big points), and goes up to `172 172`, i.e., its natural size is one inch by one inch.

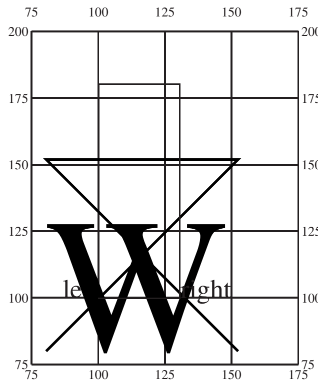
```
%! A small PostScript picture
%%BoundingBox:100 100 172 172
100 100     translate % put origin at 100 100
  0   0     moveto    % define current point
 72  72     rlineto   % trace diagonal line
 72 neg 0   rlineto   % trace horizontal line
 72  72 neg rlineto   % trace other diagonal line
stroke                % draw (stroke) the lines
  0   0     moveto    % redefine current point
/Times-Roman findfont % get Times-Roman font
72 scalefont setfont  % scale it to 1 inch
                      % make it the current font
(W) show              % draw an uppercase W
showpage              % show the page
```

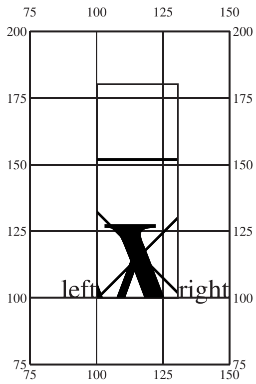
```
left\HR\framebox{\includegraphics{w.eps}}\HR right
```

left  right

In this case the picture and its bounding box nicely coincide.

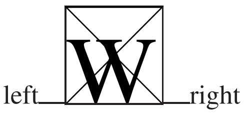
Next we are specify a bounding box which corresponds to only part of the picture, so that some parts fall outside its boundaries, and will overlay the material surrounding the picture. If the starred (clipped) form is used then the picture is clipped to the boundingbox, as shown in the case at the right. To make it easier to see what happens a grid is overlaid on the picture.



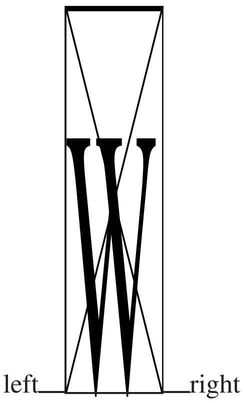


The next cases show how the `\scalebox` and `\resizebox` commands can be used in together with `\includegraphics`. Note how the image bounding box gets transformed.

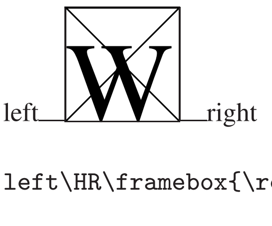
```
left\HR \framebox{\scalebox{.5}{\includegraphics{w.eps}}}\HR right
```



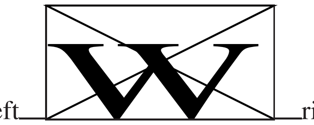

```
left\HR \framebox{\scalebox{.5}[2]{\includegraphics{w.eps}}}\HR right
```




```
left\HR \framebox{\resizebox{15mm}{!}{\includegraphics{w.eps}}}\HR right
```

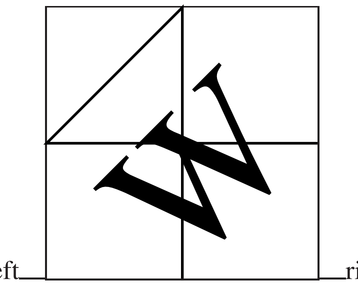



```
left\HR\framebox{\resizebox{30mm}{15mm}{\includegraphics{w.eps}}}\HR right
```

left  right

Adding rotations makes it even more fun.

```
left\HR\framebox{\rotatebox{45}{\includegraphics{w.eps}}}\HR right
```

left  right

The extended graphicx **package**

When you use the extended graphics package graphicx, the syntax is somewhat more transparent.

`\includegraphics*[`*key val list*`]{`*file*`}`

Possible keys are:

`bb`	The "bounding box" of the graphics image. Its value field should contain four dimensions, separated by spaces.
`bbllx`	Lower left x coordinate (obsolete[1]).
`bblly`	Lower right y coordinate (obsolete[1]).
`bburx`	Upper right x coordinate (obsolete[1]).
`bbury`	Upper right y coordinate (obsolete[1]).
`natheight`	Natural height of figure[2].
`natwidth`	Natural width of figure[2].
`angle`	Rotation angle (in degrees, counterclockwise).
`width`	Required width.
`height`	Required height.
`scale`	Scale factor.
`clip`	Clip the graphic to the bounding box. Can be either "`true`" or "`false`" (specifying no value is equivalent to "`true`").
`draft`	Locally switch to draft mode. Is a Boolean valued key, like "`clip`".
`type`	Specify the graphics type.
`ext`	Specify the file extension.
`read`	Specify the file extension of the file being read.

[1]Kept for backward compatibility only. `bbllx=a, bblly=b, bburx=c, bbury=d` is equivalent to `bb = a b c d`, so the `bb=` form should be used.

[2]These arguments can be used for setting the lower left coordinate to (`0 0`) and the upper right coordinate to (`natwidth natheight`).

`command` Specify any command to be applied to the file.

The starred form is just for compatibility with the standard version. It is equivalent to specifying the `clip` key.

If dimensions are given without units then big points are assumed.

The first seven keys specify the size of the image. It needs to be specified in case the file cannot be read by TeX, contains incorrect size information, or when one wishes to clip the image to a certain rectangle.

The next four keys specify scaling or rotation that should be applied to the image. Similar effects can be obtained using the standard graphics package and the `\includegraphics` command by placing the latter inside the argument of a `\resizebox`, `\rotatebox` or `\scalebox` command (see the examples above).

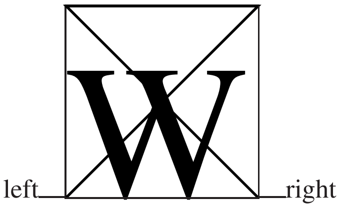
Keys are read left-to-right, so `[angle=90, height=2cm]` means rotate by 90 degrees, and then scale to a height of 2cm, while `[height=2cm, angle=90]` would result in a final *width* of 2cm.

By default LaTeX reserves for the image the space specified either in the file, or in the optional arguments. If any part of the image is actually outside this area, it will overprint the surrounding text. If the starred form is used, or the `clip` option specified, any part of the image outside this area is not printed.

The last four keys suppress the parsing of the filename. When they are used, the main *file* argument should have no file extension (see the description of `\DeclareGraphicsRule` command below).

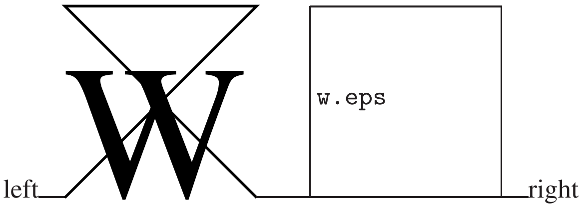
Below we repeat the examples given in the previous section, this time using the syntax of the extended graphics package. In most cases this form is easier to understand than the earlier version.

```
left\HR\framebox{\includegraphics{w.eps}}\HR right
```

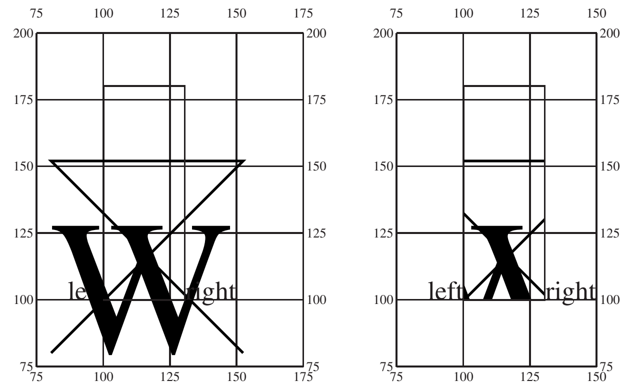


In this case the picture and its bounding box nicely coincide. When using the draft option, one only gets the bounding box.

```
left\HR\includegraphics{w.eps}\HR\HR\includegraphics[draft]{w.eps}\HR right
```

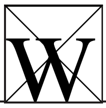


Next we only consider part of the picture, so that some parts fall outside its boundaries, and will over-print material surrounding the picture. By clipping we get rid of the "superfluous" bits.

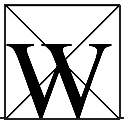


Next let us resize the picture. Note that we can only scale the picture globally in both directions.

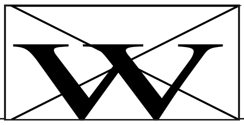
```
left\HR\framebox{\includegraphics[scale=.5]{w.eps}}\HR right
```

left  right

```
left\HR\framebox{\includegraphics[width=15mm]{w.eps}}\HR right
```

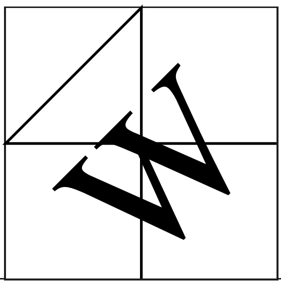
left  right

```
left\HR\framebox{\includegraphics[height=15mm,width=30mm]{w.eps}}\HR right
```

left  right

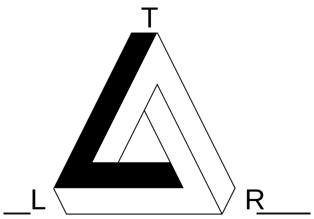
Adding rotations makes it even more fun.

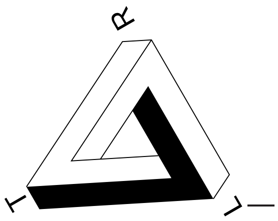
```
left\HR\framebox{\includegraphics[angle=45]{w.eps}}\HR right
```

left  right

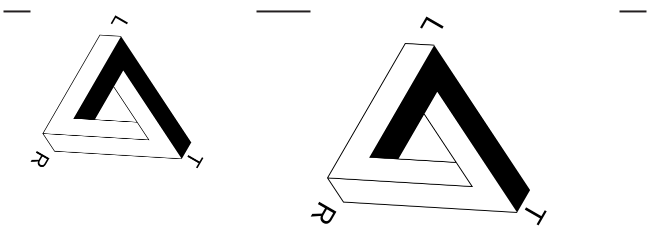
A few more examples are shown below.

```
\HR \includegraphics[width=3cm]{Escher.eps}\HR
\HR \rotatebox{-240}{\includegraphics[width=3cm]{Escher.eps}}\HR
```



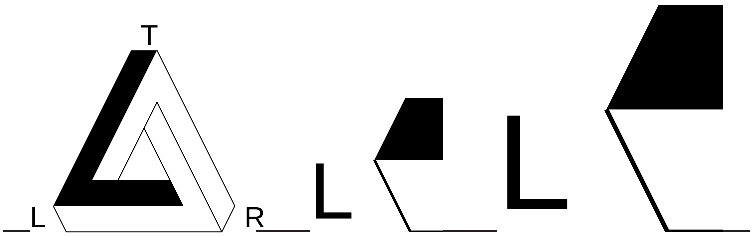



```
\HR\includegraphics[angle=240,width=30mm]{Escher.eps}\HR
\HR\includegraphics[width=30mm,angle=240]{Escher.eps}\HR
```



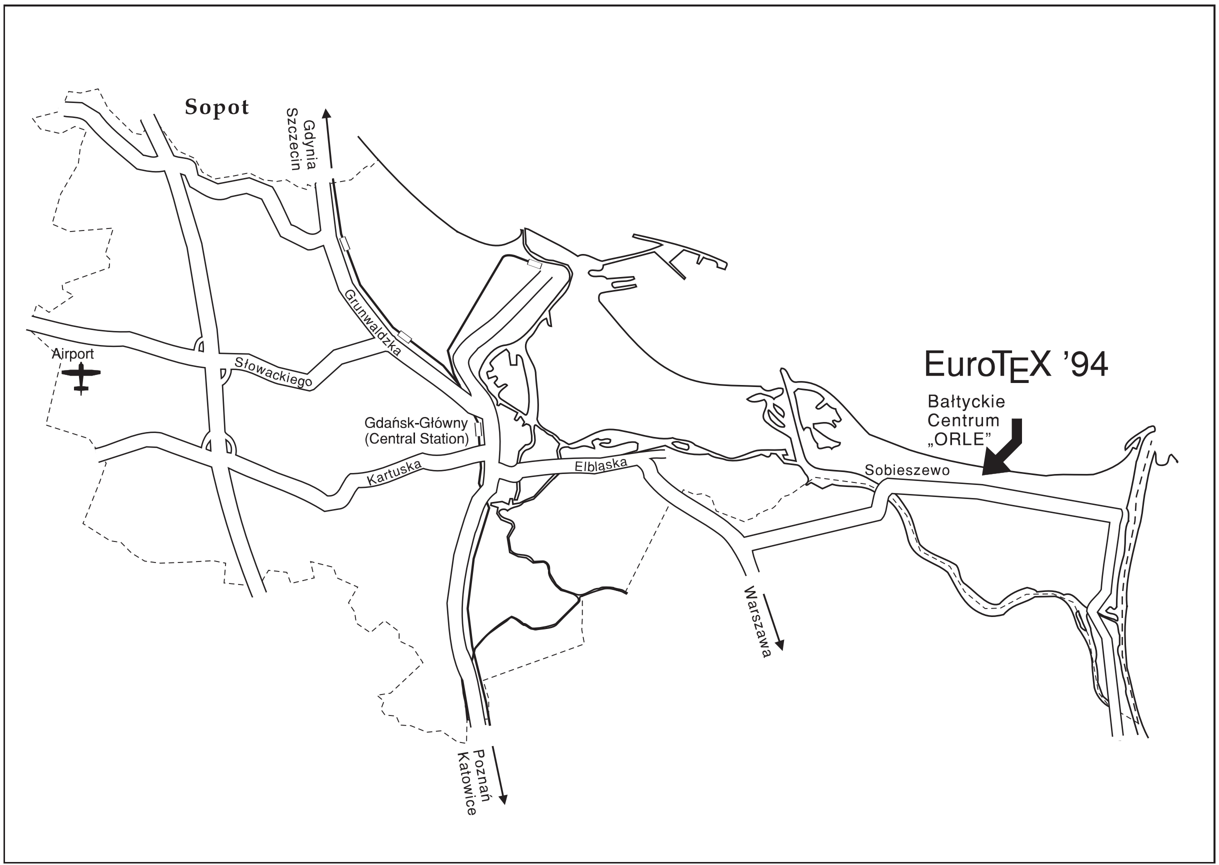
Finally as an example of the use of the bounding box one can use the above Escher picture and "cut out" part of the picture, as follows (inside the file `Escher.eps` one the native bounding box is specified as `0 0 236 236`).

```
\HR\includegraphics[width=3cm]{Escher.eps}\HR
\HR\includegraphics[bb=0 0 50 50,clip]{Escher.eps}\HR
\HR\includegraphics[bb=0 0 50 50,width=3cm,clip]{Escher.eps}\HR
```



Rotating and resizing a picture can be useful for making it fit a papersize as closely as possible. The example below shows a map of the Gdańsk area, that was made to fit an A4 page by putting it in landscape and making its width (after rotation) equal to the text width that had previously been set equal to the width of the paper size (21 cm). The `\hoffset` and `\voffset` lengths had to be reset, since LATEX's article class has its first printable point approximately at 40 mm from the upper right hand corner of the page.

```
\documentclass[a4paper]{article}
\usepackage[dvips]{graphics}
\setlength{\textwidth}{210mm}\setlength{\textheight}{297mm}
\setlength{\voffset}{-40mm}\setlength{\hoffset}{-40mm}
\pagestyle{empty}
\begin{document}
\includegraphics[angle=90,width=\linewidth]{gdansk.eps}
\end{document}
```



Setting key values globally

If you want to specify a global value for a set of keys, then you can use the `\setkeys` command defined in the keyval package (described below).

As an example let us consider the case where you would like your figure to be scaled to the width of the line. Then you could specify the following:

```
\setkeys{Gin}{width=\linewidth}
```

The first argument `Gin` of the `\setkeys` command refers to the `\includegraphics` command. The result will be that all images include with this command (when the graphicx is loaded) will be set to the desired width inside the current group or environment.

In the similar way one could specify any of the possible arguments of the `\rotatebox` command by using the `Grot` specifier, e.g.,

```
\setkeys{Grot}{origin=tc}
```

Advanced commands in the graphics package

A list of directories where LaTeX should search for graphics files can be specified by the command `\graphicspath`, whose syntax is:

`\graphicspath{`*dir-list*`}`

dir-list is a list of directories, each inside a pair of braces `{}` (even if there is only one in the list). For example, `\graphicspath{{./eps/}{./tiff/}}` causes LaTeX to look in the subdirectories `eps` and `tiff` of the current directory. By default LaTeX will look for graphics files in the same directories where it looks for other files.

`\DeclareGraphicsExtensions{`*ext-list*`}`

The `\DeclareGraphicsExtensions` command lets you specify the behaviour of the system when no file extension is given in the argument of the `\includegraphics` command. `{`*ext-list*`}` is a comma separated list of file extensions. Full filenames are constructed by appending each extension of the list *ext-list* in turn, until a file corresponding to the generated full filename is found.

Since the previous algorithm tests for the existence of a file to determine which extension to use, in the case where the `\includegraphics` command is specified without extension the graphics file must exist at the time LaTeX is run. However, if a file extension *is* specified, e.g., `\includegraphics{gr.eps}` instead of `\includegraphics{gr}`, then the graphics file need not exist at the time of the LaTeX run[3]. However, LaTeX needs to know the size of the image so that size must be specified in the arguments of the `\includegraphics` command, or in the file actually read by LaTeX (this can be the graphics file itself, or another file specified with the `read=` argument or constructed from the list of file extensions). In the latter case that file must exist at the time LaTeX is used.

With the declaration below the `\includegraphics` command will first look for a file `file.ps` and, if no such file exists, for a file `file.ps.gz`.

```
\DeclareGraphicsExtensions{.ps,.ps.gz}
  \includegraphics{file}
```

If one wants to make sure that a full filename must always be specified, then the following declaration can be used. Note also that, in the cases below, the size of the (bitmap) image is specified explicitly in each case on the `\includegraphics` command.

```
\DeclareGraphicsExtensions{{}}
  \includegraphics[1in,1in]{file.pcx}
  \includegraphics[75pt,545pt][50pt,530pt]{file.pcx}
  \includegraphics[bb=75 545 50 530]{file.pcx}
```

The action that has to take place when a file with a given extension is encountered is controlled by the following command:

`\DeclareGraphicsRule{`*ext*`}{`*type*`}{`*read-file*`}{`*cmd*`}`

Any number of these declarations can be made. The meaning of the arguments are given below.

ext
: The extension of the file. It can be specified explicitly or, if the argument to `\includegraphics` does not have an extension, from the list of extensions specified in argument *ext-list* of the `\DeclareGraphicsExtensions` command.

type
: The "type" of file involved. All files of the same type will be input with the same internal command (which must be defined in the corresponding "driver file"). For example, files with extensions `ps`, `eps`, or `ps.gz` are all classified as type `eps`.

read-file
: The extension of the file that should be read to determine the size of the graphics image. It can be identical to *ext*, but, in the case of compressed or binary images, that cannot be interpreted easily by LaTeX, the size information (the bounding box) is normally put in a separate file (e.g., for compressed gzipped PostScript files characterized by the extension `ps.gz` the corresponding readable files could have the extension `ps.bb`). If the *read-file* argument is empty `{}`, the system will not try to locate an external file to determine the size, and the latter must be specified in the arguments of *includegraphics*. If the driver file specifies a procedure for reading size files for *type*, that will be used, otherwise the procedure for reading `eps` files will be used. Therefore, in the absence of any specific other format, one can specify the size of a bitmap picture by using the syntax for PostScript images, i.e., with a `%%BoundingBox` line.

[3]For instance, it can be created on the fly by the *command* argument specified in the `\DeclareGraphicsRule` command described below.

basic PostScript							
ext	*type*	*read-file*	*cmd*	*ext*	*type*	*read-file*	*cmd*
ps	eps	ps		eps	eps	eps	
dynamic decompression							
ext	*type*	*read-file*	*cmd*	*ext*	*type*	*read-file*	*cmd*
pz	eps	bb	`‘gunzip -c #1`	ps.pz	eps	ps.bb	`‘gunzip -c #1`
eps.pz	eps	eps.bb	`‘gunzip -c #1`				
DOS related formats							
ext	*type*	*read-file*	*cmd*	*ext*	*type*	*read-file*	*cmd*
tif	tiff			pcx	bmp		
bmp	bmp			msp	bmp		
Mac related formats							
ext	*type*	*read-file*	*cmd*	*ext*	*type*	*read-file*	*cmd*
pict	pict			pntg	pntg		

Table 2.3: Arguments of `\DeclareGraphicsRule`

cmd

The command to be inserted in the `\special` command instead of the filename. In general *cmd* is empty, but for compressed files one might want to uncompress the image file before including it in the file to be printed, e.g., with the dvips driver, one could use

```
\DeclareGraphicsRule{ps.gz}{eps}{ps.bb}{‘gunzip #1}
```

where the argument `#1` denotes the full filename. In this case the final argument causes dvips to use the `gunzip` command to uncompress the file before inserting it into the PostScript output.

Possible possibilities for the various arguments of the `\DeclareGraphicsRule` command are shown in Table 2.3.

The system described so far can give some problems if the extension *ext* does not correspond to the *type* argument. You could, for instance, have a series of PostScript files called `file.1`, `file.2`, ... There is no easy way for the graphics package to automatically detect that these are PostScript files, unless a `type=eps` key setting is used for each file on the `\includegraphics` command. To make it easier to handle this situation one can set up a default type by using a `\DeclareGraphicsRule` declaration for a type `*`.

A declaration

```
\DeclareGraphicsRule{*}{eps}{*}{}
```

will cause all unknown extensions to be treated as EPS files, and the respective graphics files will be read to search for a BoundingBox comment.

2.2.5 Other packages in the graphics bundle

The epsfig and rotating packages

Two of Sebastian Rahtz's LaTeX 2.09 style files have been interfaced to the graphicx package, so that users of those styles can continue to use the syntax they are familiar with. The rotating package offers also some supplementary high level functionality, like the `sidewaystable` and `sidewaysfigure` environments.

The trig package

The trig package is not intended to be used diectly in documents. It calculates sine, cosine and tangent trigonometrical functions, that are used to calculate the space taken up by a rotated box. This package is also used by the Alan Jeffrey's fontinst program which converts PostScript files to a form useable by TeX.

The keyval package

This package is intended to be used by other packages. It implements a generic way of setting "keys" as used by the graphicx package, and splitting up the comma separated lists of *key=value* pairs. It is also used by Timothy van Zandt's pstricks package.

The lscape package

The lscape package requires and takes the same options as the graphics package. It defines a `landscape` environment within which page bodies are rotated through 90 degrees. The page head and foot are not affected, they appear in the standard (portrait) position.

2.2.6 Bézier approximations

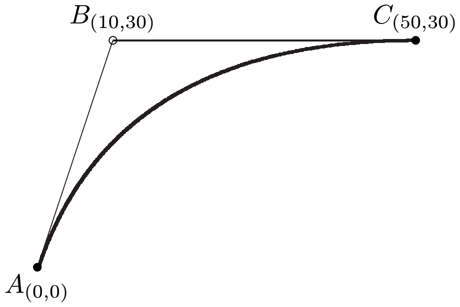
LaTeX 2_ε allows the construction of quite complicated mathematical curves using the technique of approximations with Bézier splines. Note that the PostScript language also uses (third order, or cubic) Bézier curves as the basis of its curve drawing functions.

`\qbezier[`*N*`](`*AX,AY*`)(`*BX,BY*`)(`*CX,CY*`)`

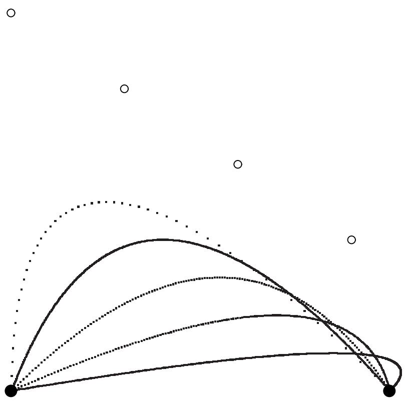
The above command defines a quadratic Bézier curve, which is defined by its two end points, (*AX,AY*) and (*CX,CY*), with (*BX,BY*) as the control point. The optional parameter *N*, if present, specifies that $N + 1$ points are plotted to approximate the curve. In the first example below A and C are the end points and B is the control point. The number of plotted points needed to obtain a smooth curve is calculated automatically (default case).

Varying the number of dots and the control point has a clear effect. In the second example two curves use the default number of points, while the others explicitly specify the number to use. The position of the control points is indicated by an open circle.

```
\setlength{\unitlength}{1mm}
\begin{picture}(55,35)(-3,-3)
  \linethickness{1pt}
  \qbezier(0,0)(10,30)(50,30)
  \thinlines
  \put(0,0){\line(1,3){10}}
  \put(50,30){\line(-1,0){40}}
  \put(0,0){\circle*{1}}
  \put(0,-1){%
    \makebox(0,0)[t]{$A_{(0,0)}$}}
  \put(10,30){\circle{1}}
  \put(10,31){%
    \makebox(0,0)[b]{$B_{(10,30)}$}}
  \put(50,30){\circle*{1}}
  \put(50,31){%
    \makebox(0,0)[b]{$C_{(50,30)}$}}
\end{picture}
```



```
\setlength{\unitlength}{.5mm}
\begin{picture}(120,100)(-5,0)
  \linethickness{.5pt}
  \qbezier [50](0,0)(0,100)(100,0)
  \qbezier     (0,0)(30,80)(100,0)
  \qbezier[150](0,0)(60,60)(100,0)
  \qbezier[200](0,0)(90,40)(100,0)
  \qbezier     (0,0)(120,20)(100,0)
% mark the end points
  \put(0,0){\circle*{3}}
  \put(100,0){\circle*{3}}
% mark the control points
  \multiput(0,100)(30,-20){5}{\circle{2}}
\end{picture}
```



2.3 Colour commands in LaTeX 2_ε

2.3.1 Introduction

LaTeX's color package implements basic colour functions. It relies on features that are not in TeX itself, but which must be supplied by the "driver" used to print the dvi file[4].

LaTeX's colour support is built around the idea of a system of *Colour Models*. The colour models supported by a driver may vary, but typically include:

rgb Red Green Blue: A comma separated list of three real numbers between 0. and 1., giving the components of the colour.

cmyk Cyan Magenta Yellow [K]Black: A comma separated list of four real numbers between 0. and 1., giving the components of the colour according to the additive model used in most printers.

gray Grey scale: a single real number between 0. (black) and 1. (grey).

named Not really a colour model, but a list of predefined colour names, to which the user can add his own definitions (with the `\definecolor` command, see below). In fact, for the `dvips` driver 64 "crayola" colours, as originally proposed by Jim Hafner in his colordvi package, are predefined. Their names (and grey equivalents) are shown in Fig. 2.1.

In principle other colour models can be easily defined (like pantone, X11, HSB, etc.). In each case the relevant code dealing with the new models has to be entered in the driver files.

2.3.2 Supported options

As indicated earlier, it is the driver that ultimately will have to paint the colours on the output medium, so you almost must select a driver. In general you can specify a default driver in the file `color.cfg` using the command:

```
\ExecuteOptions{dvips}
```

An option that can be useful is monochrome, since it turns off all colour commands, what can be useful for previewing.

[4]As with the graphics extensions discussed in section 2.2, not all drivers support the same features, and even the internal method of accessing these extensions varies between drivers. Consequently all these packages take options such as "`dvips`" to specify which driver is being used.

Figure 2.1: The predefined colours in the `named` model (grey rendering)

2.3.3 Defining Colours

The colours `black`, `white`, `red`, `green`, `blue`, `cyan`, `magenta`, `yellow` should be predefined by any driver. Packages can define other colours. If you want to define colours yourself, you can use the `\definecolor` command.

`\definecolor{`*name*`}{`*model*`}{`*colour specification*`}`

This defines *name* as a colour which can be used in later colour commands. For example

```
\DefineColor{Myorange}{cmyk}{0,0.42,1,0}
\definecolor{Myblue}{rgb}{0.8,0.85,1}
\definecolor{Mygrey}{gray}{0.75}
\definecolor{Mygreen}{named}{SpringGreen}
```

After these definitions, `Myorange`, `Myblue`, and `Mygrey` can now be used in addition to the already predefined colours, while `Mygreen` is an alias for the named colour `SprinGreen`.

2.3.4 Using Colours

The syntax for colour changes is similar to that of font changes, and comes in two forms, one declarative, and the other with arguments for local changes.

`\color[`*model*`]{`*specification*`}`
`\textcolor[`*model*`]{`*specification*`}{`*text*`}`

Without the optional argument `[`*model*`]` predefined colour names can be used as *specification*. Note that `\textcolor{name}{text}` is equivalent to `{\color{name}text}`. For instance, and example using predefined colours is:

```
{\color{green} Text starts off in green
\textcolor{red}{ a little red}
{\color{blue}nested blue text}
returning to green}
```

Text starts off in green a little red nested blue text returning to green

This is equivalent to directly specifying the colour in, e.g., the RBG model, as follows:

```
{\color[rgb]{0,1,0} Text starts off in green
\textcolor[rgb]{1,0,0}{ a little red}
{\color[rgb]{0,0,1}nested blue text}
```

Text starts off in green a little red nested blue text returning to green

Another example, mixing predefined and local colour specifications, is the following list:

```
\begin{enumerate}
\item \textcolor[rgb]{0, 1, 0}{green rgb} black
\item \color[gray]{0.5}
  \textcolor{blue}{predefined blue} grey text
\end{enumerate}
```

1. green rgb black
2. predefined blue grey

A more complex example, using the CMYK model to define a colour `Ora`, for `Orange`, also shows that the colour is defined when the box is filled with the `\sbox` command, not when it is typeset (i.e., the colour characteristics are stored with the box, and the surrounding colour does not influence the that colour).

```
\definecolor{Ora}{cmyk}{0, 0.6, 0.8, 0}
\newsavebox{\X}
\sbox{\X}{[framed black text]
 \color{Ora}[framed orange text]}
1. We \usebox{\X} return to black\\
2. {\color{Ora} it's orange
\usebox{\X} and once again orange}
```

1. We [framed black text] [framed orange text] return to black
2. it's blue now [framed black text] [framed orange text] and once again blue

2.3.5 Page Colour

`\pagecolor{`*name*`}`

The background colour of the whole page can be set using `\pagecolor`. This takes the same argument forms as `\color` but sets the background colour for the current and all subsequent pages. As it is a global declaration, you must use `\pagecolor{white}` to return to a white background (the default setting).

2.3.6 Coloured box backgrounds

Two commands similar to `\framebox` produce boxes with the backgrounds shaded with a given colour.

`\colorbox{`*background colour*`}{`*text*`}`
`\fcolorbox{`*frame colour*`}{`*background colour*`}{`*text*`}`

The examples below show how these commands are used.

```
\colorbox{red}{Black text on red background}
```

Black text on red background

```
\fcolorbox{blue}{red}{Black text, red background, blue frame}
```

Black text, red background, blue frame

```
\fcolorbox{red}{blue}{\color{white}White text, blue background, red frame}
```

White text, blue background, red frame

A combination of colour definitions and colour boxes can be used as follows:

```
\definecolor{Brown}{cmyk}{0, 0.8, 1, 0.6}
\definecolor{Yellow}{rgb}{1, 1, 0}
\definecolor{Light}{gray}{.80}
\definecolor{Dark}{gray}{.20}
\noindent\colorbox{Brown}{\textcolor{white}{White background}}\qquad
\colorbox{Yellow}{\textcolor{red}{Red background}}\\
\colorbox{Light}{\textcolor{Dark}{Light background}}\qquad
\colorbox{Dark}{\textcolor{white}{Dark background}}\\
```

White background   Red background
Light background   Dark background

A few final examples show you how the exact form of the box can be controlled.

```
\setlength{\fboxrule}{6pt}%
\colorbox{yellow}{Fun with colour}\qquad
\fcolorbox{red}{yellow}{Fun with colour}
```

Fun with colour   Fun with colour

```
\colorbox{green}{Fun with colour}\qquad
\setlength{\fboxrule}{0pt}%
\setlength{\fboxsep}{10pt}%
\colorbox{green}{Fun with colour}
```

Fun with colour   Fun with colour

ORLANDO					PRICE
Glasgow	OG4G	Thu 20/10	Mon 31/10 or 07/11	11 or 18 days	£339
		Thu 27/10	Mon 31/10 or 07/11	4 or 11 days	£339
	OG7A	Sun 23/10	Mon 31/10 or 07/11	8 or 15 days	£339
		Sun 30/10	Mon 07/11	8 days	£339

Figure 2.2: A coloured table made with the colortab package

2.3.7 A more complex example—a coloured list

Combining the use of PostScript fonts and colour one can construct lists wil colourful elements:

```
\newenvironment{coldinglist}[2]%
{\begin{list}{\textcolor{#2}{\ding{#1}}}{}}%
{\end{list}}%

\begin{coldinglist}{113}{red}
  \item Farallon's \textcolor{blue}{Replica}
  \begin{coldinglist}{42}{green}
    \item runs only on Windows;
    \item transforms PostScript type 1 fonts
          into bitmaps;
    \item well integrated in the environment.
  \end{coldinglist}
  \item \textcolor{blue}{Common Ground} of No
   Hands Software
   ....
```

❑ Farallon's Replica

- ☛ runs only on Windows;
- ☛ transforms PostScript type 1 fonts into bitmaps;
- ☛ well integrated in the environment.

❑ Common Ground of No Hands Software

- ☛ Mac and Windows;
- ☛ limited to four fixed resolutions;
- ☛ well integrated in the environment.

❑ Adobe's Acrobat

- ☛ Mac, Windows and Unix (Sun);
- ☛ handles any PostScript document with Distiller;
- ☛ does not include a Reader.

More complicated colour support can be obtained in the framework of the `pstricks` package or David Carlisle's recently released colortbl package (Figure 2.2).

The pstricks package provides a full featured interface between PostScript and LaTeX. Unfortunately the colour support in that package is slightly incompatible with the colour mechanism defined in the color package. To correct this (temporary) incompatibility the pstcol package modifies a small number of internal pstricks functions. As the pstricks package is loaded by pstcol, one can use any colours defined by color package commands within pstricks commands, and vice versa.

2.4 Supported drivers

The following driver configuration files are available, and implement part or all of the functionality of colour, and graphics, described in this section.

`dvips`	(Rokicki) full functionnality (*reference driver*)
`dvipsone`	(Y&Y) no colour stack
`dvitops`	(J. Clark) no nested rotations
`dviwin`	(H. Sendoukas) colour; file inclusion
`dviwindo`	(Y&Y) file inclusion, colour
`emtex`	(E. Mattes) file inclusion only
`pctexps`	(PCTeX) file inclusion, colour, rotation
`pctexwin`	(PCTeX) file inclusion, colour, rotation
`pctexhp`	(PCTeX) file inclusion
`oztex`	(A. Trevorrow) file inclusion only
`textures`	(Blue Sky) all functionality

Chapter 3: CERN extensions

Class files were designed for several CERN specific documents. They are available on all supported platforms, and have associated example files, which are present in the standard TEX input directories.

The `times` package used on the `\usepackage` command in the examples tells LATEX to use PostScript fonts instead of the Computer Modern font family.

3.1 Official CERN letter

The `cernlett` class produces the CERN official letter head and has four optional non-standard commands flagged as such in the example below, which is available as a file `cernlett.tex` in the nornal TEX input directories. The generated output is shown on the next page.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                             %
% Example of making an CERNletter with LaTeX                  %
%                                                             %
% Comments to Michel Goossens / goossens@cern.ch              %
%                                                             %
% Last Mod.   4 July 1995   m.g.                              %
%                                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\documentclass[12pt]{cernlett}
%\usepackage{times}
\telephone{7673363}
\ourref{xxx}                        % Non-Standard
\yourref{yyy}                       % Non-Standard
\faxdirect{7678630}                 % Non-Standard
\electronicmail{goossens@cern.ch}   % Non-Standard
\signature{M. Goossens\\CN Division}
\begin{document}
\begin{letter}{Prof. Donald E. Knuth,\\
               Computer Science Department\\
               Stanford University\\
               Stanford, CA 94305\\
               U.S.A.}
\opening{Dear Professor,}
Is it correct, that \TeX{} is the first English word in computer science?
\closing{Yours sincerely,}
\cc{All \TeX{} users}
\end{letter}
\end{document}
```

ORGANISATION EUROPÉENNE POUR LA RECHERCHE NUCLÉAIRE
EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

Laboratoire Européen pour la Physique des Particules
European Laboratory for Particle Physics

GENVE, SUISSE
GENEVA, SWITZERLAND

Adresse postale / Postal address *:

CERN
CH - 1211 GENÈVE 23

Téléphone / Telephone :
Direct +41 22 7673363
General +41 22 767 6111
Télex / Telex : 419000 CER CH
Télécopieur / Telefax :
Direct +41 22 7678630
General +41 22 767 6555
Electronic mail : goossens@cern.ch

Prof. Donald E. Knuth,
Computer Science Department
Stanford University
Stanford, CA 94305
U.S.A.

Votre référence / Your reference yyy
Notre référence / Our reference xxx

6 July 1995

Dear Professor,

Is it correct, that TEX is the first English word in computer science?

Yours sincerely,

M. Goossens
CN Division

cc: All TEX users

*Adresse pour le courrier posté en France: CERN: Site de Prévessin F-01631 CERN CEDEX

3.2 Preparing a fax

One can prepare a fax by using the cernfax class. The example below is available as a file `cernfax.tex` in the nornal TEX input directories. The generated output is shown on the next page.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                              %
% Example of making an CERN fax with LaTeX                     %
%                                                              %
% Comments to Michel Goossens / goossens@cern.ch               %
%                                                              %
% Last Mod.   4 July 1996   m.g.                               %
%                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\documentclass[12pt]{cernfax}
\usepackage{times}
\begin{document}
\begin{faxcover}
\faxfrom{783.12.34}%         Your direct fax number
\namefrom{Michel Goossens}% Your name (and title, if you want)
\addressfrom{CN Division}%  Your division, group,... (One line only)
\nameto{Prof. John Recipient}% Your correspondent
\addressto{Physics Department\\Famous University\\Famous Town\\XX1234-Famous
 Country}
\date{27 November 1999}%    To be given only if you do not like todays date
\totalpages{1}%             Total nb. of pages (incl. cover
\end{faxcover}
Dear John,

This is the start of my message to you.
As you see the message can start on
the same page as the cover.
If I want to use same larger type size, i could put the command \verb|\large|
just in front of the ‘‘Dear John", and all characters would be bigger, like
for the following text.

\large
The basis of the cernfax style is article, thus
all commands that are available for that
style can also be used here.
For instance, if I do not want a page number on the
cover page, I would use a \verb|\thispagestyle{empty}| command.

Or I could also decide to go to the next page with the
command \verb|\newpage|.

Truely yours,

Michel
\end{document}
```

EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH
ORGANISATION EUROPÉENNE POUR LA RECHERCHE NUCLÉAIRE
Laboratoire Européen pour la Physique des Particules
European Laboratory for Particle Physics

Adresse postale / Postal address:

CERN	CERN - Site de Prévessin
CH - 1211 GENÈVE 23	F - 01631 CERN CEDEX
Suisse (Switzerland)	France

Telefax cover sheet / Page de couverture de télécopie

To/À:	**Prof. John Recipient**
	Physics Department
	Famous University
	Famous Town
	XX1234-Famous Country
Fax:	
From/De:	Michel Goossens
	CN Division
Fax:	(022) 783.12.34
Phone/Tél.:	(022)
Date:	27 November 1999

Number of/Nombre de pages: 1 (cover/couverture incl.)

Dear John,

This is the start of my message to you. As you see the message can start on the same page as the cover. If I want to use same larger type size, i could put the command `\large` just in front of the "Dear John", and all characters would be bigger, like for the following text.

The basis of the cernfax style is article, thus all commands that are available for that style can also be used here. For instance, if I do not want a page number on the cover page, I would use a `\thispagestyle{empty}` command.

Or I could also decide to go to the next page with the command `\newpage`.

Truely yours,

Michel

1

3.3 Example of a CERN memorandum

An example of a CERN memorandum is shown below. A longer example is available in the standard TEX input directories in a file `cernmemo.tex`. The resulting output is shown on the next page.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                              %
% Example of a CERN memorandum with LaTeX                      %
%                                                              %
% Comments to Michel Goossens / goossens@cern.ch               %
%                                                              %
% Last Mod.   4 July 1995   m.g.                               %
%                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\documentclass[12pt]{cernmemo}
\usepackage{times}
\begin{document}
\copies{Name1 Copy1/AS, Name2-Name3 Family/CN}
\to{Fname1~Student1/AG, Fname2~Student2/AS, Fname3~Student3/AT,\\
    Fname4~Student4/CN, Fname5~Student5/ECP,Fname6~Student6/PE,\\
    Fname7~Student7/PPE,Fname8~Student8/SL  }
\from {Michel Goossens /CN-ASD}
\subject{LaTeX\ course}
\begin{memo}
Dear Colleague,

You are kindly invited to attend the course ``\LaTeX\ for Beginners''
which starts next Monday, November 5th 2999.

The aim of the course is to teach the participants (i.e. YOU) to help
yourself if you have problems and to know where to look.

Users can bring specific applications to class to discuss their
problem(s).

The course itself will be structured (approximately) as follows:

\begin{itemize}
\item {\bf Day 1}  General introduction, principles of use, simple texts.
\item {\bf Day 2}  \LaTeX\ environments.
\item {\bf Day 3}  Constructing tables.
\item {\bf Day 4}  Mathematics.
\item {\bf Day 5}  CERN specific developments and advanced features.
\end{itemize}
The tutorial will be mainly practical and a printer will be available in
the classroom. Students can bring their own documents to work on
if they so desire.

I hope you will be able to attend and look forward to meeting you
next Monday.
\end{memo}
\end{document}
```

Ref: Date: 6 July 1995

MEMORANDUM

From/De	**:**	Michel Goossens /CN-ASD
To/à	**:**	Fname1 Student1/AG, Fname2 Student2/AS, Fname3 Student3/AT, Fname4 Student4/CN, Fname5 Student5/ECP, Fname6 Student6/PE, Fname7 Student7/PPE, Fname8 Student8/SL
Subject/Sujet	**:**	LaTeX course
Copies	**:**	Name1 Copy1/AS, Name2-Name3 Family/CN

Dear Colleague,

You are kindly invited to attend the course "LaTeX for Beginners" which starts next Monday, November 5th 2999.

The aim of the course is to teach the participants (i.e. YOU) to help yourself if you have problems and to know where to look.

Users can bring specific applications to class to discuss their problem(s).

The course itself will be structured (approximately) as follows:

- **Day 1** General introduction, principles of use, simple texts.
- **Day 2** LaTeX environments.
- **Day 3** Constructing tables.
- **Day 4** Mathematics.
- **Day 5** CERN specific developments and advanced features.

The tutorial will be mainly practical and a printer will be available in the classroom. Students can bring their own documents to work on if they so desire.

I hope you will be able to attend and look forward to meeting you next Monday.

3.4 Example of making minutes of a meeting

An example of CERN minutes is shown below. A longer example is available in the standard TEX input directories in a file `cernmins.tex`. The resulting output is shown on the next page.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                    %
% Example of making  MINUTES  for a meeting with LaTeX               %
%                                                                    %
% Comments to Michel Goossens / goossens@cern.ch                     %
%                                                                    %
% Last Mod.  June 13  1995  m.g.                                     %
%                                                                    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\documentclass[12pt]{cernmins}
\usepackage{times}
\begin{document}
\Title{Minutes of the HEP/CCC of Monday June 10th 1991}
\Stitle{CERN/CCC/91-6}
\Those{\bf Present}{P.~Borgeaud (for M.~Banner), M.~Froissart,
                    H.~Hoffmann, W.~Hoogland (chairman), C.M.~Jones (secretary),
                    G.~Kalmus, E.~Remiddi (for N.~Cabibbo), F.~Roehner (for H.E.~Stier),
                    F.~Sacton, J.J.~Thresher, R.~Tirler (for M.~Banner), P.~Zanella}
\Those{\bf Excused}{S.~Loken}
\Those{\bf Invited for specific points\ }{%
R.K.~B\"ock, R.~Brun, B.~Carpenter, J.S.~Hutton, L.M.~Robertson}
\begin{enumerate}
\item Minutes of the last meeting (CCC/CERN/86/04)
\begin{itemize}
\item The Chairman welcomed the new member G. Kalmus from RAL, and the
  new secretary C.M.S.  Jones from CERN. He thanked H. Grote for his
  sterling work as former secretary.
\end{itemize}
\item Matters arising from the minutes
\begin{itemize}
\item Several members queried the state of the GKS-3D contract
  negotiations; who is writing which drivers for which machines, at
  what cost and on what timescale.  The relevant specialists not being
  present at the meeting, it was agreed to attach this information to
  the minutes. {\bf Action: CMJ}
\item \ldots
\end{itemize}
\item \ldots
\item Date of the next meeting
\par The next meeting will be held on {\bf Monday 16 March 1987} at 09:00
in the Director General's conference room in the CERN main building.
\begin{center}Recorder: C.M.S. Jones\end{center}
\end{enumerate}
\end{document}
```

Minutes of the HEP/CCC of Monday June 10th 1991

Present: P. Borgeaud (for M. Banner), M. Froissart, H. Hoffmann, W. Hoogland (chairman), C.M. Jones (secretary), G. Kalmus, E. Remiddi (for N. Cabibbo), F. Roehner (for H.E. Stier), F. Sacton, J.J. Thresher, R. Tirler (for M. Banner), P. Zanella

Excused: S. Loken

Invited for specific points : R.K. Böck, R. Brun, B. Carpenter, J.S. Hutton, L.M. Robertson

1. Minutes of the last meeting (CCC/CERN/86/04)
   - The Chairman welcomed the new member G. Kalmus from RAL, and the new secretary C.M.S. Jones from CERN. He thanked H. Grote for his sterling work as former secretary.
2. Matters arising from the minutes
   - Several members queried the state of the GKS-3D contract negotiations; who is writing which drivers for which machines, at what cost and on what timescale. The relevant specialists not being present at the meeting, it was agreed to attach this information to the minutes. **Action: CMJ**
   - …
3. …
4. Date of the next meeting

   The next meeting will be held on **Monday 16 March 1987** at 09:00 in the Director General's conference room in the CERN main building.

Recorder: C.M.S. Jones

3.5 Example of making the agenda for a meeting

An example for making the agenda for a CERN meeting is shown below. It is available in the standard TEX input directories in a file `cernagen.tex`. The resulting output is shown on the next page.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                    %
% Example of making an AGENDA for a meeting with LaTeX               %
%                                                                    %
% Comments to Michel Goossens / goossens@cern.ch                     %
%                                                                    %
% Last Mod.  June 13  1995  m.g.                                     %
%                                                                    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\documentclass[12pt]{cernmins}
\usepackage{times}
\begin{document}
\Title{Agenda for the HEP/CCC Meeting}
\begin{flushleft}
{\bf Time:} Monday June 10th 1991 at 9:00\\
{\bf Place:}
Director General's conference room, sixth floor, Main building.
\end{flushleft}
\Those{\bf Invited}{Members of the CCC, L.M.~Robertson,
B.~Carpenter,\\ J.S.~Hutton, R.~B\"ock, R.~Brun, A.~Berglund.}
\begin{itemize}
\item Minutes of the last meeting (CCC/CERN/86/04)
\item Matters arising from the minutes
\item Report on MEDDLE meetings
\item \ldots
\item Date of the next meeting
\end{itemize}
\begin{center}
\bf Secretary: C.M.S. Jones
\end{center}
\end{document}
```

Agenda for the HEP/CCC Meeting

Time: Monday June 10th 1991 at 9:00
Place: Director General's conference room, sixth floor, Main building.

Invited: Members of the CCC, L.M. Robertson, B. Carpenter,
J.S. Hutton, R. Böck, R. Brun, A. Berglund.

- Minutes of the last meeting (CCC/CERN/86/04)
- Matters arising from the minutes
- Report on MEDDLE meetings
- …
- Date of the next meeting

Secretary: C.M.S. Jones

3.6 Example of a CERN preprint class

A preprint class `cernart.cls` with option `12pt` corresponding to the format of preprints, which historically were composed in the "typing pool" of CERN, has been developed. Users can study the example `crnartex.tex` in the standard TEX input directories. This style is quite flexible, to be able to cope with the many types of title pages (from 1 to > 500 authors, coming from few to many institutes ...). This style should be used for **all** preprints at CERN. For compatibility reasons one can also typeset preprints at text sizes of 10 and 11 points by specifying the options `10pt`, resp. `11pt`.

It is extremely important to construct your preprint like the example shown, particularly the titlepage, as the central library automatically extracts bibliographic information from the source of your preprint. This can only be done if the authors and institutes, for example, are correctly tagged with the commands `\authlist`, `\Iref` and `\Instfoot`. On request of the library and the publishers, a few commands were added to mark up supplementary information on a title page – `\submitted`, `\collaboration`, `\conference`, `\note` and `\dedicated`. They produce one or more lines of italized centered text at the current vertical location on the output page. The user can specify them in any order and control the amount of white space by adding commands like `\vspace{-5mm}`.

Two forms of a CERN title page layout are shown on the next two pages. The first (Fig. 3.1) is used for a small number of authors, while the second (Fig. 3.2) can be used to cope with a large number of authors and institutes (e.g. the LEP experiments). An agreement has been reached with North Holland Publishing Company, that articles, which are written in "standard" LATEX and use the proposed markup will be handled within 24 hours.

If you wish to convert your article to Microsoft Word, this can be done, provided you stick to the format of the example.

```
\documentclass[12pt]{cernart}
\usepackage{times}
\begin{document}
\begin{titlepage}
\docnum{CERN--PPE/91--49}
\date{7 March 1991}
\title{TEST-BEAM PERFORMANCE OF A TRACKING TRD PROTOTYPE}
\begin{Authlist}
J.T.~Shank, J.S.~Whitaker
\Instfoot{a1}{Boston University, Boston, Mass., USA.}
V.A.~Polychronakos,  V.~Radeka,  D.~Stephani
\Instfoot{a2}{Brookhaven National Laboratory, Upton, NY, USA.}
H.~Beker,  R.K.~Bock,  M.~Botlo,  L.~DiCiaccio\Aref{c},
M.~Koratzinos,  A.~Lopez\Aref{*}
\Instfoot{a3}{CERN, Geneva, Switzerland.}
W.J.~Willis
\Instfoot{a4}{Columbia University, New York, NY, USA.}
T.~Akesson
\Instfoot{a5}{Lund University, Lund, Sweden.}
V.~Chernyatin,  B.~Dolgoshein,  P.~Nevski
\Instfoot{a6}{Moscow Physical Engineering Institute, Moscow, USSR.}
\end{Authlist}
\collaboration{Our collaboration}
\begin{abstract}
This sample article shows how to type a CERN PPE preprint using the
standard generic notation for the tagging of authors, institutes and
the likes on the title page. It also teaches how to
include a picture and a table and how to get the typographic type
correct right inside mathematics mode. Furthermore there is a
citation and a bibliography. This ends the funny abstract.
\end{abstract}
\vspace{2cm}% Lower the following text somewhat
\submitted{(Submitted to some important Journal)}
\conference{To be presented by the author\\
on the conference in somewhere at sometimes.}
\note{NOTE: not yet final}
\dedication{Dedicated to AAA}
\Anotfoot{c}{Univ. di Roma II, ‘Tor Vergata’, Rome, Italy.}
\Anotfoot{*}{On leave from Fac. de Ciencias F\’\i sicas,
Univ. Complutense, Madrid, Spain.}
\end{titlepage}
...
\end{document}
```

EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

CERN–PPE/91–49
7 March 1991

TEST-BEAM PERFORMANCE OF A TRACKING TRD PROTOTYPE

J.T. Shank, J.S. Whitaker
Boston University, Boston, Mass., USA.

V.A. Polychronakos, V. Radeka, D. Stephani
Brookhaven National Laboratory, Upton, NY, USA.

H. Beker, R.K. Bock, M. Botlo, L. DiCiaccio[c)], M. Koratzinos, A. Lopez[*)]
CERN, Geneva, Switzerland.

W.J. Willis
Columbia University, New York, NY, USA.

T. Akesson
Lund University, Lund, Sweden.

V. Chernyatin, B. Dolgoshein, P. Nevski
Moscow Physical Engineering Institute, Moscow, USSR.

Our collaboration

Abstract

This sample article shows how to type a CERN PPE preprint using the standard generic notation for the tagging of authors, institutes and the likes on the title page. It also teaches how to include a picture and a table and how to get the typographic type correct right inside mathematics mode. Furthermore there is a citation and a bibliography. This ends the funny abstract.

(Submitted to some important Journal)

To be presented by the author
on the conference in somewhere at sometimes.

NOTE: not yet final

Dedicated to AAA

[c)] Univ. di Roma II, ‘Tor Vergata’, Rome, Italy.
[*)] On leave from Fac. de Ciencias Físicas, Univ. Complutense, Madrid, Spain.

Figure 3.1: CERN preprint title page for few authors

```
\documentclass[12pt]{cernart}
\usepackage{times}
\begin{document}
\begin{titlepage}
\docnum{CERN--PPE/91--49}
\date{7 March 1991}
\title{TEST-BEAM PERFORMANCE OF A TRACKING TRD PROTOTYPE}
\begin{Authlist}
J.T.~Shank\Iref{a1}, J.S.~Whitaker\Iref{a1},V.A.~Polychronakos\Iref{a2},
V.~Radeka\Iref{a2}, D.~Stephani\Iref{a2}, H.~Beker\Iref{a3},
R.K.~Bock\Iref{a3}, M.~Botlo\Iref{a3}, W.J.~Willis\Iref{a4},
T.~Akesson\Iref{a5}, V.~Chernyatin\Iref{a6}, B.~Dolgoshein\Iref{a6},
P.~Nevski\Iref{a6}, L.~DiCiaccio\IAref{a3}{c},
M.~Koratzinos\IIIref{a3}{b1}{b2} and A.~Lopez\IIAref{a3}{b2}{*}
\end{Authlist}
\begin{abstract}
This sample article shows how to type a CERN PPE preprint using the
standard generic notation for the tagging of authors, institutes and
the likes on the title page. It also teaches how to
include a picture and a table and how to get the typographic type
correct right inside mathematics mode. Furthermore there is a
citation and a bibliography. This ends the funny abstract.
\end{abstract}
\vspace{2cm}% Lower the following text somewhat
%
\submitted{(Submitted to some important Journal)}
%
\Instfoot{a1}{Boston University, Boston, Mass., USA.}
\Instfoot{a2}{Brookhaven National Laboratory, Upton, NY, USA.}
\Instfoot{a3}{CERN, Geneva, Switzerland.}
\Instfoot{a4}{Columbia University, New York, NY, USA.}
\Instfoot{a5}{Lund University, Lund, Sweden.}
\Instfoot{a6}{Moscow Physical Engineering Institute, Moscow, USSR.}
\Instfoot{b1}{Blackett Lab., Imperial College, London, UK.}
\Instfoot{b2}{LAL, IN2P3--CNRS and Univ. Paris-Sud, Orsay, France.}
\Anotfoot{c}{Univ. di Roma II, ‘Tor Vergata’, Rome, Italy.}
\Anotfoot{*}{On leave from Fac. de Ciencias F\'\i sicas,
Univ. Complutense, Madrid, Spain.}
\end{titlepage}
...
\end{document}
```

EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

CERN–PPE/91–49
7 March 1991

TEST-BEAM PERFORMANCE OF A TRACKING TRD PROTOTYPE

J.T. Shank[1)], J.S. Whitaker[1)],V.A. Polychronakos[2)], V. Radeka[2)], D. Stephani[2)], H. Beker[3)], R.K. Bock[3)], M. Botlo[3)], W.J. Willis[4)], T. Akesson[5)], V. Chernyatin[6)], B. Dolgoshein[6)], P. Nevski[6)], L. DiCiaccio[3,c)], M. Koratzinos[3,7,8)] and A. Lopez[3,8,*)]

Abstract

This sample article shows how to type a CERN PPE preprint using the standard generic notation for the tagging of authors, institutes and the likes on the title page. It also teaches how to include a picture and a table and how to get the typographic type correct right inside mathematics mode. Furthermore there is a citation and a bibliography. This ends the funny abstract.

(Submitted to some important Journal)

---

1) Boston University, Boston, Mass., USA.
2) Brookhaven National Laboratory, Upton, NY, USA.
3) CERN, Geneva, Switzerland.
4) Columbia University, New York, NY, USA.
5) Lund University, Lund, Sweden.
6) Moscow Physical Engineering Institute, Moscow, USSR.
7) Blackett Lab., Imperial College, London, UK.
8) LAL, IN2P3–CNRS and Univ. Paris-Sud, Orsay, France.
c) Univ. di Roma II, ‘Tor Vergata’, Rome, Italy.
*) On leave from Fac. de Ciencias Físicas, Univ. Complutense, Madrid, Spain.

Figure 3.2: CERN preprint title page for many authors

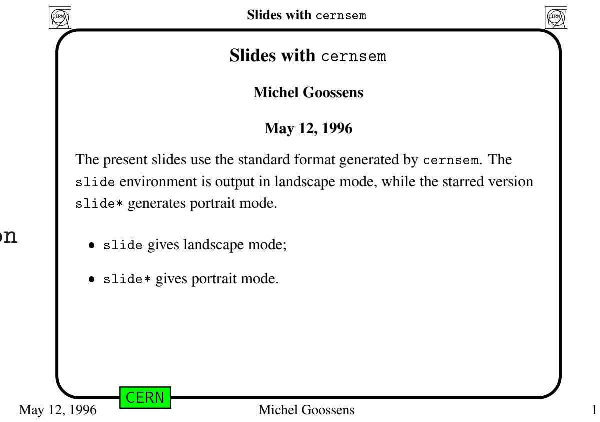
3.7 Making CERN slides with the `cernsem` class

The class file cernsem, based on the seminar package (see Section 5), has all the advantages of seminar, but provides a simpler and CERN-specific user interface.

For making slides it suffices to specify the `cernsem` class and use the `slide` environment, which, by default, generates slides in landscape mode.

```
\documentclass[dvips]{cernsem}
\title{Slides with \texttt{cernsem}}
\author{Michel Goossens}
\begin{document}
\begin{slide}
 \maketitle
 The present slides use the standard
 format generated by \texttt{cernsem}.
 The \texttt{slide} environment is output
 in landscape mode, while the starred version
 \texttt{slide*} generates portrait mode.
 \begin{itemize}
  \item \texttt{slide} gives landscape mode;
  \item \texttt{slide*} gives portrait mode.
 \end{itemize}
\end{slide}
```

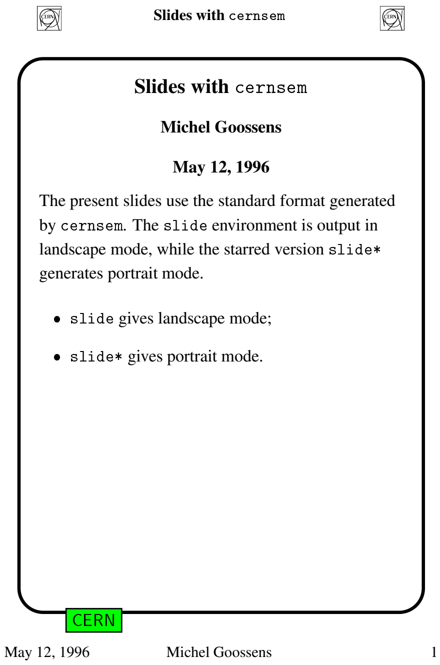
This gives the following output:



If you prefer portrait mode, then you can use the `slide*` environment. Moreover, if most or all slides are in portrait mode it is useful to specify the `portrait` option on the `\documentclass` command to show them upright with ghostview.

```
\documentclass[dvips,portrait]{cernsem}
\title{Slides with \texttt{cernsem}}
\author{Michel Goossens}
\begin{document}
\begin{slide*}
 \maketitle
 The present slides use the standard
 format generated by \texttt{cernsem}.
 The \texttt{slide} environment is output
 in landscape mode, while the starred version
 \texttt{slide*} generates portrait mode.
 \begin{itemize}
  \item \texttt{slide} gives landscape mode;
  \item \texttt{slide*} gives portrait mode.
 \end{itemize}
\end{slide*}
```

This gives the output shown at the right:



You can easily add a title on each slide with the commands

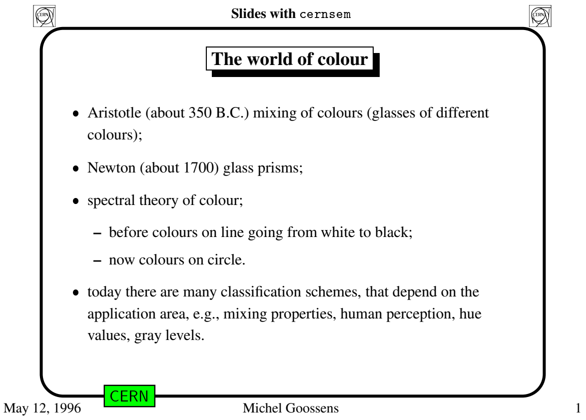
`\Slide{`*Title*`}`	`\PSlide{`*Title*`}`

where `\Slide` and `\PSlide` generate landscape and portrait layout, respectively.

An example in landscape mode:

```
\begin{Slide}{The world of colour}
\begin{itemize}
 ...
\end{itemize}
\end{Slide}
```

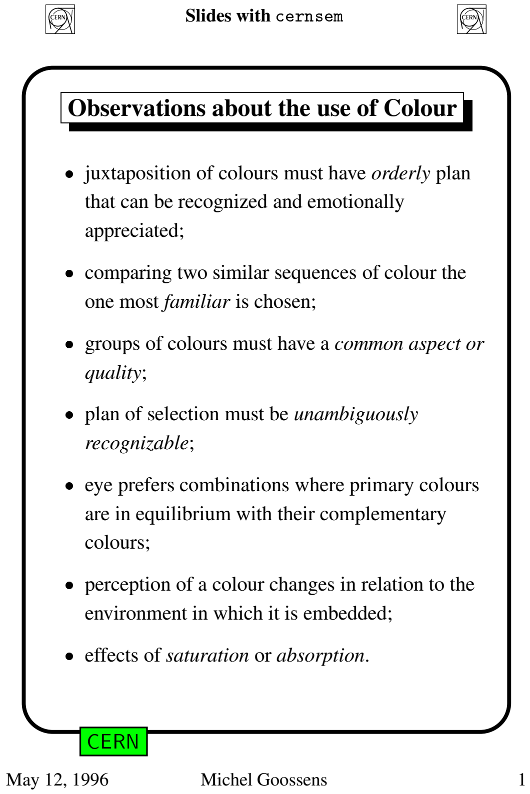
would give the following:



An example in portrait mode:

```
\begin{PSlide}{Observations about the use
               of Colour}
\begin{itemize}
 ...
\end{itemize}
\end{Slide}
```

gives as output:



Colour can be introduced with the command

`\SlideColours{`*text-col*`}{`*back-col*`}`

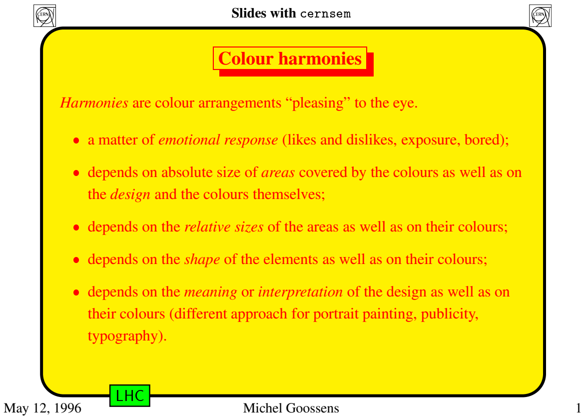
where *text-col* is the colour of the text, and *back-col* the background colour. Note also that the text on the border of the slide ("CERN" by default) can be changed by redefining the string `\Semlabel`, as follows:

```
\renewcommand{\Semlabel}{my_string}
```

as shown below, where we chose the word "LHC".

```
\renewcommand{\Semlabel}{LHC}
\SlideColours{red}{yellow}
\begin{Slide}{Colour harmonies}
\emph{Harmonies}  are colour arrangements
‘‘pleasing’’ to the eye.
\begin{itemize}
 ...
\end{itemize}
\end{Slide}
```

The output is shown at the right (greyscale simulation):



You can use all colours defined in LaTeX’s `color` package, (see CNL 221, the article on colour at the URL `http://wwwcn.cern.ch/cnls/221/node65.html` or as a PostScipt file at `ftp://asisftp.cern.ch/cnl/221/textproc.ps.gz`). Remember to be a little careful with colour by trying not to use too many of them, since an overdose of colours dilutes the message one wants to get across. In many cases just grayscales, or one or two highlight colours suffice to emphasize part of the text.

Slides can be further customized (other running titles, fonts, denser text, etc.), but to do this you will have to read carefully the article in CNL 213 or the basic documentation of the seminar package, which is available as a PostScript file on Unix as `/usr/local/doc/TeX/seminar.ps`. Such customization is most easily put in the configuration file `cernsem.cfg`, which is read automatically by the `cernsem` class.

```
% Example of cernsem.cfg file for an LHC meeting
\newpagestyle{LHC}
  {{\color{Black}\small\raisebox{-1ex}[0mm][0mm]{\makebox[0mm][l]{\huge LHC}}
                 \hfil {\bfseries \@title} \hfil \thepage}}%
  {{\footnotesize\today \hfil\@author\hfil LHC Meeting}}
\slideframe{plain}
\pagestyle{LHC}
\endinput
```

Chapter 4: Preparing texts

4.1 Recommendations for scientific texts

In scientific texts the printed form of a symbol often implies a meaning which is not easily captured by generic markup. Therefore authors using some form of generic coding (like LaTeX or SGML) need to know about typographical conventions.

4.1.1 Summary of typographic conventions for scientific texts

The following is a brief summary of the most important rules for composing scientific texts [15, 27, 28].

1. The most important rule is **consistency**: a symbol should always be the same, whether is appears in a formula or in the text, on the main line or as a superscript or subscript. This means that in TeX, once a symbol has been defined in mathematics mode ('$'), it should always be used in math mode. Note that inside math mode, TeX by default prints characters in *italics*.

   For scientific work, however, quite a few symbols must be set in **roman** (upright) characters[1]. This is the case for the following families of symbols, which represent the names of:

   – units, such as g, cm, s, keV. Note that physical constants are usually in italics, so units involving constants are mixed roman-italics, e.g. GeV/*c* (where the *c* is italic because it symbolizes the speed of light, a constant);
   – particles, for example p, K, q, H. For elementary particles the PEN (Particle Entity Notation) scheme is proposed (see the URL `http://wwwcn.cern.ch/asdoc/WWW/pennameswww/pennameswww.html`, the file `/usr/local/share/texmf/doc/latex/pennames/pennames.ps` or Appendix B.2);
   – standard mathematical functions (sin, det, cos, tan, Re, Im, etc.). Use the built-in LaTeX functions for these (`\sin` etc.);
   – chemical elements, for example Ne, O, Cu;
   – numbers;
   – names of waves or states (p-wave) and covariant couplings (A for axial, V for vector), names of monopoles (E for electric, M for magnetic);
   – abbreviations that are initials of bits of words (exp, for experimental; min, for minimum);
   – the 'd' in integrands (e.g. d*p*).

2. Let the text processor do as much work as it can. Do not try to change the defaults too much; this will decrease the portability and maintainability of your documents. TeX implements a lot of the rules mentioned above by default in math mode.

3. Do not add blanks at random to make formulae look "nicer'‘.

4. Refrain from using specific page layout commands (like `\newline` with LaTeX). You will forget that you put them in your text and later wonder why some text is badly adjusted or starts a new line.

It is important for the clarity of the text that these rules, especially those in point 1, be followed in all cases, since it will help the reader understand at first glance what the author means. Some significant examples in which the correct type (italic or upright) of the symbol changes its meaning are shown below:

[1] With LaTeX, roman type in math mode can be produced by the `\mathrm` command.

roman type		italic type	
A	ampere (electric unit)	A	atomic number (variable)
e	electron (particle name)	e	electron charge (constant)
g	gluon (particle name)	g	gravitational constant
l	litre (volume unit)	l	length (variable)
m	metre (length unit)	m	mass (variable)
p	proton (particle name)	p	momentum (variable)
q	quark (particle name)	q	electric charge (variable)
s	second (time unit)	s	c.m. energy squared (variable)
t	tonne (weight unit)	t	time (variable)
V	volt (electric unit)	V	volume (variable)
Z	Z boson (particle name)	Z	atomic charge (variable)

4.1.2 Entity definitions for elementary particles

In texts on high energy physics, frequently re-occurring strings are the names of elementary particles. For example, the Z^0 particle can be coded in various different ways with LaTeX: `$\mbox{Z}^0$`, `$\mathrm{Z^0}$` and `Z$^0$` all achieve the same typographical effect, a roman Z with a superscript 0. In the interest of standardization and typing convenience, the "Particle Entity Naming" scheme has been proposed, as emntioned above. It not only relieves the user from having to worry about the correctness of what he types, but also will allow an automatic extraction of the particle names from the input file, so that it will be easy to enter data about an article using this convention into a database of abstracts (see Appendix B.2). A few examples follow.

```
\newcommand{\PWm}{\mbox{$\mathrm{W}^-$}}  \newcommand{\PWp}{\mbox{$\mathrm{W}^+$}}
\newcommand{\PZz}{\mbox{$\mathrm{Z}^0$}}
\newcommand{\PMW}{\mbox{$m_{\mathrm{W}}$}}\newcommand{\PMZ}{\mbox{$m_{\mathrm{Z}^0}$}}

The masses of the \PWm, \PWp\ and \PZz\ particles are
\PMW\ and \PMZ\ respectively. In general one has $\PMW > \PMZ$.
```

This yields the following output (apart from the box, which is only used for clarity!):

The masses of the W^-, W^+ and Z^0 particles are m_{W} and m_{Z^0} respectively. In general one has $m_{\mathrm{W}} > m_{\mathrm{Z}^0}$.

It is seen that the commands operate as expected both in math and in text mode.

4.2 Interesting document classes and packages

This section describes classes or packages and utility programs, which are not or little described in the LaTeX book. First there are a few extensions made by Lamport himself, then a brief description is given of other style-files, which have been copied to the TeX file base at CERN.

4.2.1 The `proc/ieee` Classes

The proc class is based on the article document class. It produces two-column output for ACM and IEEE conference proceedings. The command `\copyrightspace` makes the blank space at the bottom of the first column of the first page, where the proceedings editor will insert a copyright notice. This command works by producing a blank footnote, so it is placed in the text of the first column. It must go after any `\footnote` command that generates a footnote in that column.

LaTeX automatically numbers the output pages. It's a good idea to identify the paper on each page of output. Placing the command `\markright{Jones---Foo}` in the preamble (before the `\begin{document}` command) prints "Jones—Foo" at the bottom of each page.

4.2.2 AMS – American Mathematical Society

The complete functionality of $\mathcal{A}\!\mathcal{M}\!\mathcal{S}$-TeX is now available with the amstex package of LaTeX, whose facilities are described in documents that can be found via the Web interface at the URL `http://wwwcn.cern.ch/dci/asis/products/TeX/index.html`.

Why would a LaTeX user want to bother with $\mathcal{A}\!\mathcal{M}\!\mathcal{S}$-LaTeX?

If your writing contains a significant proportion of mathematics, and you care about the quality of the printed results, then sooner or later you'll find shortcomings in standard LaTeX and want to remedy them. Chances are that at least the first few of the shortcomings you encounter will be ones that are already addressed by an $\mathcal{A}\!\mathcal{M}\!\mathcal{S}$-LaTeX package. If you want to have maximum mathematical typesetting power ready at hand, rather than stop to cast about for a solution whenever you run into some unusual demand in your writing, then $\mathcal{A}\!\mathcal{M}\!\mathcal{S}$-LaTeX will go a long way toward meeting your needs.

If you are a long-time LaTeX user and have lots of mathematics in what you write, then you may recognize solutions for some familiar problems in this list of $\mathcal{A}\!\mathcal{M}\!\mathcal{S}$-LaTeX features:

- A convenient way to define new 'operator name' commands analogous to `\sin` and `\lim`, including proper side spacing and automatic selection of the correct font style and size (even when used in sub- or superscripts).
- Multiple substitutes for the `eqnarray` environment to make various kinds of equation arrangements easier to write.
- Equation numbers automatically adjust up or down to avoid overprinting on the equation contents (unlike `eqnarray`).
- Spacing around equals signs matches the normal spacing in the `equation` environment (unlike `eqnarray`).
- A way to produce multiline subscripts as are often used with summation or product symbols.
- An easy way to substitute a variant equation number for a given equation instead of the automatically supplied number.
- An easy way to produce subordinate equation numbers of the form (1.3a) (1.3b) (1.3c) for selected groups of equations.
- A `\boldsymbol` command for printing bold versions of individual symbols, including things like ∞ and lowercase Greek letters.
- An amsthm package that provides a useful `proof` environment and some enhancements to the `\newtheorem` command: support for multiple theorem styles in a single document and for unnumbered theorem types.

Using an AMS package in a LaTeX document

Many of the principal features of $\mathcal{A}\!\mathcal{M}\!\mathcal{S}$-LaTeX are provided in separate packages so that they can be used individually on demand. The amsmath package is perhaps the single most noteworthy package, as it subsumes the amstext, amsbsy, and amsopn packages, and provides a number of other enhancements for mathematical typesetting. The current list of packages is:

amsmath Defines extra environments for multiline displayed equations, as well as a number of other enhancements for math.

amstext Provides a `\text` command for typesetting a fragment of text inside a display.

amsbsy Defines `\boldsymbol` and `\pmb` 'poor man's bold' commands.

amsopn	Provides `\DeclareMathOperator` for defining new 'operator names' like `\sin` and `\lim`.
amsthm	Provides a `proof` environment and extensions for the `\newtheorem` command.
amsintx	Provides more descriptive command syntax for integrals and sums.
amscd	Provides a `CD` environment for simple commutative diagrams (no support for diagonal arrows).
amsxtra	Provides certain odds and ends such as `\fracwithdelims` and `\accentedsymbol`.
upref	Makes `\ref` print cross-reference numbers always in an upright/roman font regardless of context.

The amsmath package has the following options:

centertags	(default) For a split equation, place equation numbers vertically centered on the total height of the equation.
tbtags	'Top-or-bottom tags': For a split equation, place equation numbers level with the last (resp. first) line, if numbers are on the right (resp. left).
sumlimits	(default) Place the subscripts and superscripts of summation symbols above and below, in displayed equations. This option also affects other symbols of the same type—$\prod$, $\coprod$, $\bigotimes$, $\bigoplus$, and so forth—but excluding integrals (see below).
nosumlimits	(default) Always place the subscripts and superscripts of summation-type symbols to the side, even in displayed equations.
intlimits	Like sumlimits, but for integral symbols.
nointlimits	(default) Opposite of intlimits.
namelimits	(default) Like sumlimits, but for certain 'operator names' such as $\det$, $\inf$, $\lim$, $\max$, $\min$, that traditionally have subscripts placed underneath when they occur in a displayed equation.
nonamelimits	Opposite of namelimits.

The amsmath package also recognizes the following options which are normally selected (implicitly or explicitly) through the `\documentclass` command, and thus need not be repeated in the option list of the `\usepackage{amsmath}` statement.

leqno	Place equation numbers on the left.
reqno	Place equation numbers on the right.
fleqn	Position equations at a fixed indent from the left margin rather than centered in the text column.

Recommended use of math font commands

If you find yourself employing math font commands frequently in your document, you might wish that they had shorter names, such as `\mb` instead of `\mathbf`. Of course, there is nothing to keep you from providing such abbreviations for yourself by suitable `\newcommand` statements. But for LaTeX to provide shorter names would actually be a disservice to authors, as that would obscure a much better alternative: defining custom command names derived from the names of the underlying mathematical objects, rather than from the names of the fonts used to distinguish the objects. For example, if you are using bold to indicate vectors, then you will be better served in the long run if you define a 'vector' command instead of a "math-bold" command:

```
\newcommand{\Vek}[1]{\mathbf{#1}}
```

whereupon you can write `\VeK{a} + \Vek{b}` to produce $\mathbf{a} + \mathbf{k}$. If you decide several months down the road that you want to use the bold font for some other purpose, and mark vectors by a small over-arrow instead, then you can put the change into effect merely by changing the definition of `\Vek`; otherwise you would have to replace all occurrences of `\mathbf` throughout your document, perhaps even needing to inspect each one to see whether it is indeed an instance of a vector. The table below shows a list of font commands in math mode available with the amstex package.

`\mathbb`	Blackboard bold alphabet. Only uppercase is available. $\mathbb{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$
`\boldsymbol`	Used to obtain bold numbers and other nonalphabetic symbols, as well as bold Greek letters. $\boldsymbol{\infty\ \pi\ \omega\ \diamond \in \oplus \nabla \Rightarrow}$
`\mathcal`	Calligraphic letters. Only uppercase is available. $\mathcal{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$
`\mathrm`	Roman, normal shape. Note that most of the time `\text` can be used instead. $\mathrm{abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ}$
`\mathit`	Math italic (font is automatically selected in math mode, so `\mathit` is normally not needed. $\mathit{abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ}$
`\mathsf`	Math sans serif. $\mathsf{abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ}$
`\mathbf`	Math bold. $\mathbf{abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ}$
`\mathtt`	Math teletype. $\mathtt{abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ}$
`\pmb`	"Poor man's bold," for math symbols when bold versions are not available in the fonts. $\pmb{abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ}$

4.2.3 Elsevier publishing

These are LaTeX packages for Nuclear Physics A and Physics Letters. See the URL `http://wwwcn.cern.ch/dci/asis/products/TeX/latex/elsevier/instraut.ps.gz` for more details.

4.2.4 APS - American Physics Society Journals

The series of journals edited by the American Physics Society, like *Physical Review*, *Physical Review Letters* and *Reviews of Modern Physics* allow the use of LaTeX with the revtex style file. In the directory `/usr/local/share/texmf/doc/latex/revtex` you can find more information. In particular the files `manintro.ps.gz` is a compressed PostScript file describing the principles of operation of this family os files. The same directory also contains a set of sample TeX source files.

Please note that currently these styles *only* work in LaTeX 2.09 mode, so you *cannot* use LaTeX 2$_\varepsilon$ commands with that package and you *must* use the old syntax `\documentstyle[..]{revtex}`.

4.2.5 Drawing Feynman diagrams

A package for drawing Feynman diagrams based on LaTeX's `picture` environment, was developed by Michael Levine in 1988, when he was working in the Cavendish Laboratory (Cambridge U.K.). Documentation is available as a PostScript file `feynmandoc.ps.gz` in the directory `/usr/local/share/texmf/doc/latex/feynman`. It corresponds to Michael's PhD. thesis and the LaTeX source is also available in that directory. The master file to be included in your LaTeX source is `feynman.tex`

More recently Thorsten Ohl (TH Darmstadt, Theoretical High Energy Physics Group) has developed a package feynmf (feynmp) for drawing Feynman fiagrams with LaTeX and METAFONT (METAPOST). Documentation is available at the URL `http://wwwcn.cern.ch/asdoc/WWW/feynmf.html`.

4.2.6 LaTeX packages

A large collection of LaTeX packages and classes are available on the TeX system at CERN. Information for most of them is available via the URL `http://wwwcn.cern.ch/dci/asis/products/TeX/index.html` (or in the directory `/usr/local/share/texmf/doc/latex`). You can also consult [2] which describes over 100 packages.

4.3 Supplementary utilities

4.3.1 Using `makeindex`

The LaTeX `\index` command only enters information into the `.idx` file. A separate program is needed for sorting this information and putting it in printable form, to be included later in the LaTeX document. One such program is `makeindex`, described in [1, 2], the file `/usr/local/doc/TeX/makeindex.ps.Z` or in the man file. The syntax of the index entries recognized by `makeindex` is shown in the following table,

<table>
<tr><th>Markup</th><th>Description</th></tr>
<tr><td><code>\index{entry}</code></td><td>Simple one-level entry</td></tr>
<tr><td><code>\index{text lev1!text lev2</code></td><td>Two level entry</td></tr>
<tr><td><code>\index{text lev1!text lev2!text lev3}</code></td><td>Three level entry (maximum three levels allowed)</td></tr>
<tr><td><code>\index{entry|(}</code></td><td>Start of range for <code>entry</code></td></tr>
<tr><td><code>\index{entry|)}</code></td><td>End of range for <code>entry</code></td></tr>
<tr><td><code>\index{lexical@actual}</code></td><td>Sorts as <code>lexical</code>, prints as <code>actual</code></td></tr>
<tr><td><code>\index{entry|see{other entry}}</code></td><td>Gives “entry, see other entry”</td></tr>
<tr><td><code>\index{user"@computer}</code></td><td><code>"</code> escapes special <code>\index</code> characters <code>|</code>, <code>@</code> and <code>!</code></td></tr>
</table>

4.3.2 Using BibTeX

BibTeX is a program for compiling a reference list for a document from a bibliographic database. It is run by typing

```
bibtex myfile
```

where `myfile.tex` is the name of your LaTeX input file. This reads the file `myfile.aux`, which was generated when you ran LaTeX on `myfile.tex`, and produces the file `myfile.bbl`. BibTeX should be run from the directory containing `myfile.tex` (which should be the same directory from which LaTeX was run on that file). Do not forget that you will have to run LaTeX at least once (and most probably twice) again, to get the citations correct in your document.

If the `.bib` file is not in the same directory as the LaTeX input file—for example, if you're using someone else's `.bib` file—then you must include the full path as part of the file name specified by the `\bibliography` command. On Unix a `~` cannot appear in the argument of a `\bibliography` command, so you should use a complete path name. For example, the LaTeX command

```
\bibliography{/udir/jones/bibfiles/gnus}
```

specifies the file `gnus.bib` kept by Jones in his `/bibfiles` directory.

There is now no formal provision for sharing bibliographic database information, nor are there programs to assist in making your own `bib` files. For more details, see [1, 2], the URL `http://wwwcn.cern.ch/dci/asis/products/TeX/bibtex/btxdoc.dvi.gz` or in the man file.

4.4 Accessing information on CTAN

This section briefly describes the Internet TEX archives, which form a collaborating network of archives known as CTAN (Comprehensive TEX Archive Network). Its aim is to provide easy access to TEX-related material for as many TEX users as possible. Following an initial implementation on `ftp.tex.ac.uk` in July 1992, At present the main CTAN archives are the Cambridge (UK) and Dante (Germany) Internet sites (`ftp.tex.ac.uk` and `ftp.dante.de`). They adopt the same directory structure and have identical files. In many countries sites mirror part or the complete archive locally.

The CTAN archives all run an enhanced *ftp* server, which supports dynamic compression, uncompression, and archive creation options. Fetch the top-level file `README.archive-features` for information. The server also supports site-defined commands to assist you. Please read `README.site-commands` for a brief overview.

You can go to the top of the archive tree using the alias `CTAN:`—type `cd ctan:` (*with* the colon).

If you are using LATEX and want to know whether a style file with the functionality you need exists somewhere, you can consult the WWW catalogue at the URL `http://www.dit.csiro.au/~gjw/texpkgs.html`. It is maintained by Graham Williams (in Australia). You can also find the catalogues in the CTAN directory `help/Catalogue` Its scope includes all macros that are available via anonymous ftp. The contents of the CTAN archives is also available on CD-Rom (eg, from the German TEX User Group, DANTE or from Walnut Creek.

When you want to connect to one of the ctan sites, please bear in mind the geographical location of the site to which you intend to ftp. Choose a site as close to you as possible. Also, transfer rate will be drastically improved if you transfer large files outside normal working hours (at the remote site!). The short list below shows the main TEX archives in Europe and the United States of America.

America

`ftp.tug.org`	Complete secondary CTAN mirror site in the USA.
`ftp.cdrom.com`	Complete secondary CTAN mirror site in the USA.

Europe

`ftp.dante.de`	Primary CTAN site in Germany.
`ftp.tex.ac.uk`	Primary CTAN site in Great Britain.
`ftp.loria.fr`	Complete secondary CTAN mirror site in France.

Figure 4.1 represents a WWW interface to CTAN, while at the bottom of this page an ftp session to one of the CTAN archives is shown. We want to retrieve the Catalogue file, and the `xspace` package. After connecting to the site, we first put the working directory to the ctan root by issuing a command `cd ctan:` (*note the colon*). The catalogue is in the `help/Catalogue` directory, so we go there and transfer the file. We go back to the ctan root (`cd ctan:`) and search the file base to locate the utility xspace by giving the command `quote site index xspace`, which finds two occurrences. We go to the directory where that file resides with the command `cd macros/latex/packages/tools` and transfer the file in question using `get xspace.dtx`. The ctan archives support dynamic compression and uncompression of data sets, so that a (set of) large files can be conveniently transferred. If your machine does not have uncompression tools, a compressed file can be uncompressed on the fly. The exact procedure of how these facilities work is explained in the online documentation at the ctan sites. You can also connect to the same DANTE site and search for a file via a WWW interface. The URL is `http://www.dante.de/cgi-_bin/ctan-_index`

```
$ ftp ftp.dante.de
Connected to sun.dante.de.
220 sun.dante.de FTP server (Version wu-2.4(8) Fri Jan 3 11:27:21 MET 1997) ready.
Name (ftp.dante.de:goossens): ftp
331 Guest login ok, send your complete e-mail address as password.
```

```
Password: firstname.surname@cern.ch

           ....  (Many lines of helpful information) ....

ftp> cd ctan:
250-Welcome to the Comprehensive TeX Archive Network (CTAN)...
ftp> help/Catalogue
250 CWD command successful.
ftp> dir catalogue.html
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
-rw-rw-r--   1 Mr.Ftp   server    218297 Jan 14 20:56 catalogue.html
226 Transfer complete.
ftp> quote site index xspace
200-index xspace
200-NOTE. This index shows at most 20 lines. for a full list of files,
200-retrieve /tex-archive/FILES.byname
200-1997/01/22 |       3495 | macros/latex/packages/tools/xspace.dtx
200-1995/07/13 |       2249 | systems/msdos/4alltex/disk04/xspace.zip
200  (end of 'index xspace')
ftp> cd ctan:
250 CWD command successful.
ftp> cd macros/latex/packages/tools
250 CWD command successful.
ftp> get xspace.dtx
200 PORT command successful.
150 Opening BINARY mode data connection for xspace.dtx (3495 bytes).
226 Transfer complete.
3495 bytes received in 0.33 seconds (10.21 Kbytes/s)
ftp> quit
221 Goodbye.
```

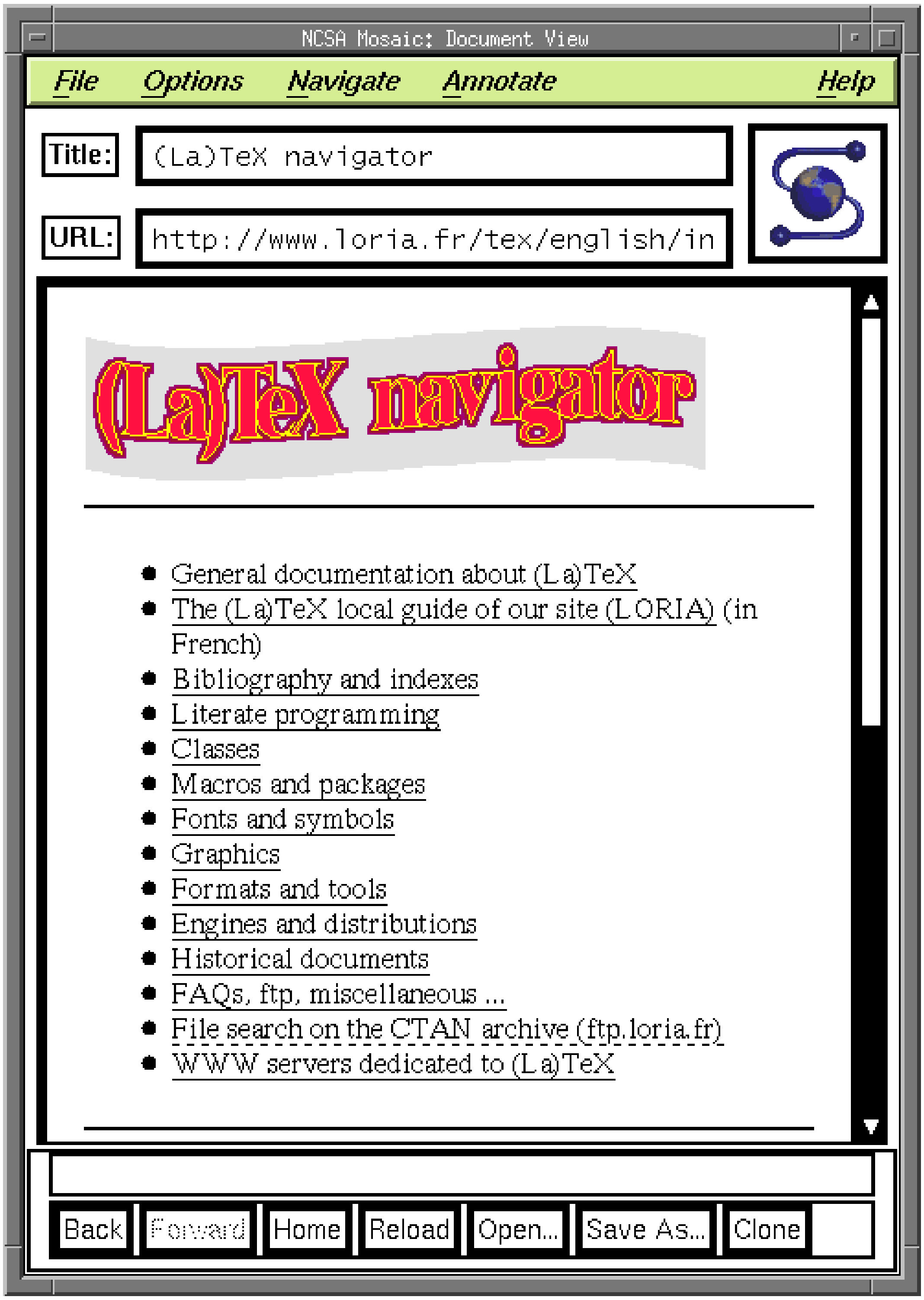


Figure 4.1: Introducing the LaTeX Navigator

A more user-friendly interface to CTAN is the three-lingual (French, English, German) LaTeX Navigator on WWW at the URL `http://www.loria.fr/tex`. One can easily locate files by using the file-search facility on this server (see penultimate line of list)

Chapter 5: Colour Slides with LaTeX and seminar

5.1 Slides and LaTeX

Many LaTeX users want to take advantage of TeX's high-quality typesetting when they produce overhead slides for a presentation. This facility was originally provided by a separate package, SLiTeX, but that had a number of disadvantages:

- it was limited to a set of specially-scaled Computer Modern fonts and it was not easy to adapt to other fonts;
- the user was required to have two separate files, one for control information and the other for the actual slides;
- the control of colour and overlays was limited and crude;
- There was only one 'style' for slides, and writing a different layout (to, say, put a logo on each slide) was not documented.

LaTeX users now have a variety of fonts, and vast numbers of styles, to choose from, but SLiTeX has lagged behind. With LaTeX 2_ε it is included as a normal class file, which emulates SLiTeX without the overhead of a separate macro package. However, there exists a much better LaTeX class file, namely seminar. If used in conjunction with a PostScript printer, and a set of useful macros called PStricks, this offers almost every imaginable facility, including:

☞ Fancy frames, headers and footers;

☞ Landscape and portrait slides in the same document;

☞ Coloured text and tables;

☞ Interleaving of annotations and slides;

☞ Slide 'chapters' and list of slides;

☞ Overlays.

`seminar` is a normal LaTeX class and can be used with almost all other LaTeX packages (such as those to change text or math fonts, include graphics, etc.). Its main job is to produce transparencies, but it can also make accompanying notes from the same file.

5.2 Using the `seminar` class

Usage is simple; begin your document in the normal way with

```
\documentclass[... options ...]{seminar}
\usepackage{... packages ...}
```

The slide environments are

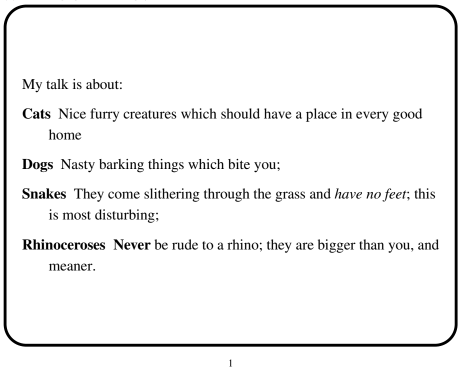
```
\begin{slide}     ...     \end{slide}

\begin{slide*}    ...     \end{slide*}
```

Where `slide` is for landscape slides and `slide*` is for portrait slides. By default, the document is typeset in *landscape* mode, but if you include the `portrait` style option, the document is typeset in portrait mode. *Typesetting* the document in landscape mode is different from *printing* it in landscape mode; you have to worry about the orientation of the page when printing, but with `dvips` this is simple, and taken care of in the local control file described below.

So the default output[1] from this input:

```
\documentclass[a4paper]{seminar}
\usepackage{times}
\begin{document}
\begin{slide}
My talk is about:
\begin{description}
\item[Cats] Nice furry creatures which
 should have a place in every good home
\item[Dogs] Nasty barking things which
 bite you;
\item[Snakes] They come slithering through
 the grass and {\em have no feet}; this is
 most disturbing;
\item[Rhinoceroses] {\bf Never} be rude to a
 rhino; they are bigger than you, and meaner.
\end{description}
\end{slide}
\end{document}
```

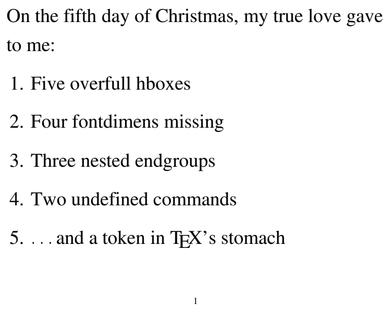
will look like:



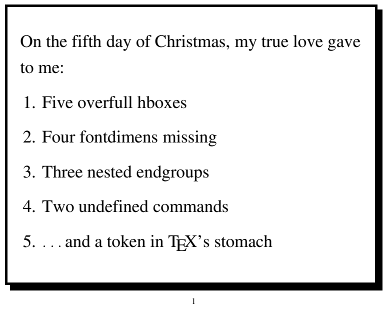
Most slides will be no more complicated than this, using standard LaTeX environments like `itemize`, `enumerate` and `tabular`.

5.3 Frame styles

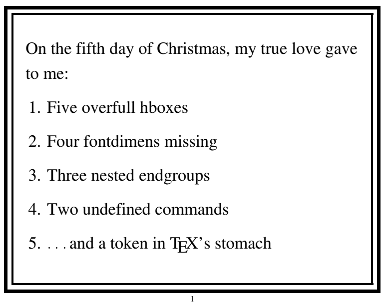
A variety of slide framing styles are available, set with the `\framestyle` command; the following are some of the predefined ones (some assume you have a PostScript printer), using the `\slideframe` command:



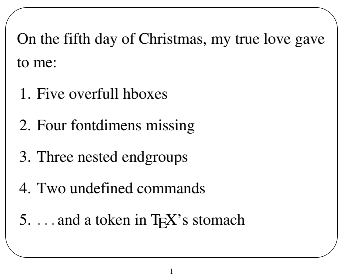
none



shadow



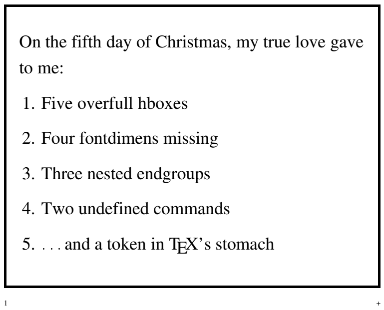
double

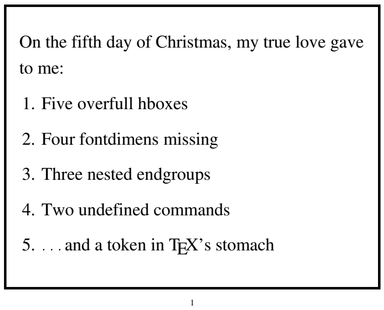


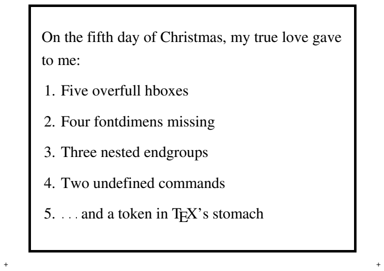
oval

[1] We have added a style option output "`times`" so that the output will reduce properly to thumbnails for this article.

Similarly, a variety of page styles (the headers and footers) are available with the `\pagestyle` command, such as:

empty 

plain 

align 

Both slide frames and page styles can be customized.

5.4 Interleaving notes, and selecting subsets

It is easy to intersperse your slides with notes to yourself; these can be simply placed between the `slide` environments or enclosed in a specific `note` environment. You can use any LaTeX commands in these notes, and include your whole article here if desired. When you want to print the slides, a variety of style options can be used:

`slidesonly` Only the slides are printed;

`notesonly` Only the slides are printed;

`notes` The slides are interleaved with the notes;

`article` The document notes are typeset like a normal LaTeX paper, and the slides are placed as figures (reduced to half size).

The `\slideplacement` command can be used to affect how slides are placed in the `\article` format; the possible parameters are:

`float` (default) Slides are floated

`float*` Slides are floated, but if two column format is chosen they will span both columns

`here` Slides are placed where they occur in the notes

Further detailed control of the interaction between slides and notes is given in the *User's Manual*.

Selected slides can be included or excluded with the `\onlyslides` or `\noteslides` commands which a parameter of a comma-separated list of slides; this can be numbers, ranges (e.g., 5–10) or LaTeX `\ref` commands referring to `\label` commands in the slides.

5.5 Control over slide size, fonts and magnification

There are a great number of parameters by which the user can change any of the following either on a slide-by-slide basis, or for the whole document:

- Slide height and width;
- Top, bottom, left and right margins;
- Text justification (it is ragged right by default);
- Page breaking by varying tolerance of over-running material;
- Inter-line spacing;
- Point size, and choice of fonts.

How to change the default settings is explained in detail in the *User's Guide*.

Because `seminar` works by magnifying pages, sophisticated users should read the manual to see how to deal with setting and changing TEX dimensions. Most users need not worry about this—in commands like `\includegraphics` you should always express your 'width' and 'height' requests in fractions of the line size anyway.

5.6 The local control file

The `seminar` package always starts by trying to find a file called `seminar.con` on the `TEXINPUTS` path; this gives the user or site an opportunity to conveniently overwrite the defaults. The `seminar.con` file can contain any LATEX commands, including inputting style files. Figure 5.1 on page 85 were typeset using a `seminar.con` set up for CERN; the contents of this are given below, with explanation of what is being done:

First, we set up landscape macros for the `dvips` driver.

```
\newcommand{\printlandscape}{%
      \special{papersize=297mm,210mm}}
```

We will assume PostScript printers, and gain nice PostScript effects like rounded box corners; these will need some extra style files:

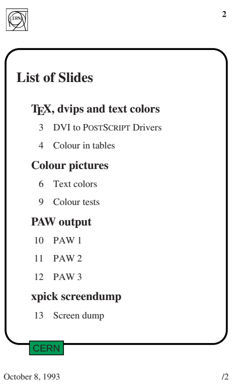
```
\input semcolor.sty
\input fancybox.sty
```

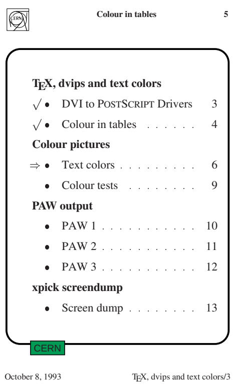
For slide 'sections', list of contents, we use another style file:

```
\input slidesec.sty
```

This allows us to use various commands, some of which are discussed below.

`\listofslides` `\Slidecontents`





For slide headings, there is a predefined `\slideheading` command; we will amend this so that it is typeset with a 'shadow'. The `slidechapter` command is also defined (the code is not given here) which allows the user to break the slides into groups; the slide chapter title will be given in the bottom right corner with this CERN style.

```
\def\@@empty{}
\renewcommand{\makeslideheading}[1]{%
\gdef\theslideheading{#1}%
\def\@@foo{#1}%
\ifx\@@foo\@@empty\else
\begin{Sbox}
\bgroup
                 \large\bfseries\begin{Bcenter}#1\end{Bcenter}%
                \egroup
                \end{Sbox}
             \centerline{\shadowbox{\TheSbox}}
             \vspace{1ex minus 1ex}
        \fi
\SlideFront
}
```

Now the CERN page and frame styles; the plain ‘cern’ style just places registration ‘+’ marks, and the date:

```
\newpagestyle{cern}%
{{\color{Black}\small{\bfseries +} \hfil \today \hfil  {\bfseries+}}}
{{\color{Black}\small{\bfseries +} \hfil \thepage\hfil {\bfseries+}}}%
```

Whereas the ‘cernsections’ style has section headings and a logo:

```
\newpagestyle{cernsections}%
{{\color{Black}\small
% call the logo procedure; the unit seems to be about 48pt
\raisebox{-2ex}[0mm][0mm]{\makebox[0mm]{\special{" .8 .8 -.5 -.3 logocrn}}}
 \hfil {\bfseries \theslideheading} \hfil {\bfseries\thepage} }}%
{{\small\color{Black}\today \hfil\thechapterheading /\inchap}}
```

For the slide frames, we define a frame with the word “CERN” set in a coloured box on the lower left; this is done using the PStricks macros (which are automatically included by the ‘semcolor’ option above). The colour commands are also defined using PStricks (the code is not listed here); the names are shown on one of the colour figures.

```
\newslideframe{cern}{{\SlideFront\boxput(-0.7,-1.11){\psframebox[%
      linecolor=Black,fillcolor=ForestGreen,fillstyle=solid]{%
      \mbox{{\normalsize\sffamily\color{Black} CERN}}}}{#1}%
\color{Black}}}
```

Finally we make sure that each slide starts with the current foreground colour.

```
\def\everyslide{\SlideFront}
\def\theslideheading{}
```

The user gives the command `\SlideColours`, with two parameters, which are colour names for foreground and background. A synonym is defined for black on white.

```
\newslideframe{blueframe}[\psset{linecolor=NavyBlue,linewidth=\slideframewidth,%
  framesep=\slideframesep,cornersize=absolute,linearc=.5cm}]{\psframebox{#1}}
\newcommand{\SlideColours}[2]{%
 \gdef\SlideFront{\color{#1}}%
 \slideframe{\Framedefault}%
 \slideframe*[\psset{fillcolor=#2,fillstyle=solid}]{blueframe}%
}
\newcommand{\blackandwhite}{\SlideColours{Black}{White}}
```

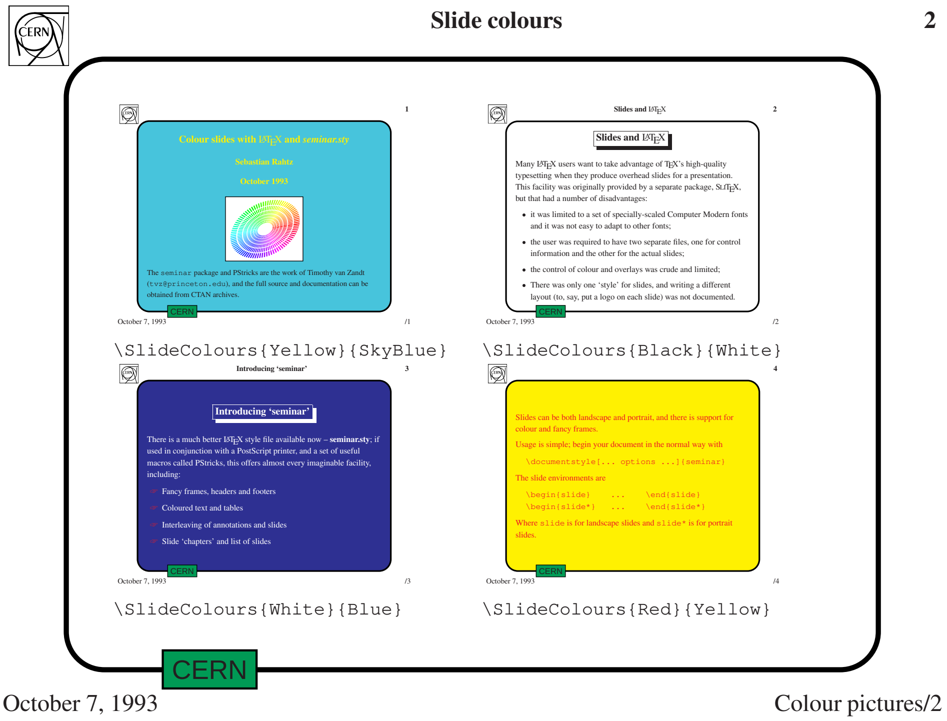


Figure 5.1: Colours slides: (simulated in black and white)

To make it easier to make landscape and portrait slides with headings to new environments `Slide` and `PSlide` have been defined.

```
% Landscape version
\newenvironment{Slide}[1]{\begin{slide}\slideheading{#1}\centering}{\end{slide}}
%
% Portrait version
\newenvironment{PSlide}[1]{\begin{slide*}\slideheading{#1}\centering}{\end{slide*}}
```

The slide defaults will be for a detailed layout and CERN logo, with yellow writing on a blue background:

```
\pagestyle{cernsections}
\slideframe{cern}
\def\Framedefault{cern}
\SlideColours{Yellow}{RoyalBlue}
```

Chapter 6: Modifying class and package files

6.1 introductory remarks

This chapter is for people who have written document styles for LaTeX 2.09 and want to upgrade them for LaTeX 2_ε. For a description of the new features of the user level commands, see chapter 2. The details about the interface for class and package writers can be found in *LaTeX 2_ε for class and package writers* (in the file `clsguide.tex`). The way LaTeX now deals with fonts is described in *LaTeX 2_ε font selection* (in the file `fntguide.tex`). Both these files come with the basic LaTeX distribution.

6.1.1 Command names

LaTeX 2_ε comes with a new set of commands. Those LaTeX users who have written their own extensions to LaTeX in the past know that in version 2.09 basically two types of commands existed, namely “internal” commands—with ‘`@`’-signs in their name—and “user level” commands—without ‘`@`’-signs in their name.

LaTeX 2_ε has also commands that have both upper- and lowercase letters in their name. Those commands are part of the interface for package and class writers. They are not intended for use in documents, but they are meant to provide an ‘easy’ interface to some of the internals of LaTeX 2_ε.

6.1.2 Filenames

LaTeX 2_ε introduces a number of new file extensions. This makes it easy to distinguish between files that contain a Document Class, files that contain an external option to a Document Class and files that contain Packages. In table 6.1 you can find an overview of the extensions that have been introduced. I would suggest that you would stick to the same set of extensions when you upgrade your old `.sty` files.

6.2 Upgrading existing ‘styles’—general remarks

6.2.1 Is it a class or a package?

The first thing to do when you upgrade an existing style file for LaTeX 2_ε, is to decide whether it should become a document class or a package. Here are a few points which might help you to decide what to do with your `.sty` file.

- Was the original `.sty` file a documentstyle? Then turn it into a document class.

`cls`	A file containing a document class
`clo`	A file containing an external option to a document class
`sty`	A file that contains (part of) a package
`cfg`	An optional file that is looked for at runtime and which can contain customization code
`def`	A file containing definitions that will be read in at runtime.
`ltx`	A file used when building the LaTeX 2_ε format
`dtx`	Documented source code for `.cls`, `.clo`, `.sty`, `.cfg`, `.def`, and `.ltx` files
`fd`	A font definition file
`fdd`	Documented source code for `.fd` files
`ins`	DOCSTRIP instructions to unpack `.dtx` and `.fdd` files

Table 6.1: Extensions for LaTeX 2_ε files

- Was the original `.sty` file meant to be used for a certain type of document? In that case you should consider turning it into a document class, possibly by building on top of an existing class. An example of this is `proc.sty` which is now `proc.cls`.
- Was it just changing some aspects of the way LaTeX does things? In that case you would probably want to turn your `.sty` file into a package.
- Was it adding completely new functionality to LaTeX? Examples of this kind of `.sty` file are files such as fancyheadings`.sty` and xypic`.sty`. This you most certainly will want to turn into a package for LaTeX 2ε.

6.2.2 Style options → packages

Trying it out unchanged

After you've decided to produce a package file, you should first try to run a document that uses your `.sty` file through LaTeX 2ε unmodified. This assumes that you have a suitable test set that tests all functionality provided by the `.sty` file. (If you haven't, now is the time to make one!) The experience of the last months has shown that most of the available `.sty` files will run with LaTeX 2ε without any modification. Yet if it does run, please enter a note into the file that you have checked that it runs and resubmit it to the archives if it was a distributed file.

Bits that might have failed

Some `.sty` files will need modification before they can be used successfully with LaTeX 2ε. Such a modification is needed for instance when you used an internal macro from the old font selection scheme. An example is `\fivrm` which is used by some packages to get a small dot for plotting. The obvious solution for this seems be to include a definition such as:

```
\newcommand{\fivrm}{\normalfont\fontsize{5}{6.5pt}\selectfont}
```

But that involves a lot of internal processing and may result in long processing times for your documents that use this. For this purpose the command `\DeclareFixedFont` is available. It bypasses a lot of the overhead of the font selection scheme. Using this command the solution becomes:

```
\DeclareFixedFont{\fivrm}{OT1}{cmr}{m}{n}{5}
```

This tells LaTeX that the command `\fivrm` should select a font with OT1 encoding, `cmr` family, medium weight, normal shape and size 5 point.

Pieces of code that might need checking

If your `.sty` file uses commands that used to be part of the way LaTeX used to deal with fonts than your file will almost certainly *not* work. You will have to look in *LaTeX 2ε font selection* or [2] to find out the details about what needs to be done.

Commands such as `\tenrm` have to be replaced by `\fontsize{10}{12pt}\rmfamily` and `\twlsf` by `\fontsize{12}{14.5pt}\sff`
Another possibility is to use the rawfonts package, described in *LaTeX 2ε for Authors*.

Also commands such as `\xipt` no longer exist and should be replaced, for instance in the case of `\xipt` by `\fontsize{11}{13.6pt}\selectfont`.

LaTeX 2.09 used commands with names beginning with `\p` for 'protected' commands. For example, `\LaTeX` was defined to be `\protect\pLaTeX`, and `\pLaTeX` produced the LaTeX logo. This made `\LaTeX` robust, even though `\pLaTeX` was not. These commands have now been reimplemented using `\DeclareRobustCommand` (described in *LaTeX 2ε for class and package writers*). If your package redefined one of the `\p`-commands, you should replace the redefinition by one using `\DeclareRobustCommand`.

When you use internal commands from NFSS version 1 you will have to be very careful to check if everything still works as it was once intended.

Note that macros such as `\rm` are now defined in class files, so their behaviour may differ for each class. Instead you should use the lower level commands such as `\rmfamily` in packages. When you want to make sure that you get a certain font, independent of the environment in which your macro is activated, you can first call `\normalfont` and then switch the various parameters of the font selection scheme as necessary.

In some cases you may need to use the user level commands such as `\textrm`. This is necessary for instance when you define a command that may also be used in mathmode.

6.2.3 Document styles → Classes

Minimal updates are necessary

When you are upgrading a document style to a document class there are a few things that you really *have* to change, or your class will not work.

One of the things that must be done, is making sure that your class doesn't define `\@normalsize` but `\normalsize`. Make sure that `\renewcommand` is used to redefine `\normalsize` as it is already defined in the kernel of LaTeX, but to produce a warning that it needs to be given a real definition.

Another aspect that needs to be dealt with, is that the parameters `\@maxsep`, `\@dblmaxsep` and `\footheight` no longer exist. The first two were part of the float placement algorithm, but a change in that algorithm made them superfluous. The parameter `\footheight` was reserved in LaTeX 2.09, but it was never used.

The declarative font changing commands (`\rm`, `\sf` etc.) are no longer defined by default. Their definitions have been moved to the class files. Make sure that you define them or that they are not used by the users of your class. The standard document classes all contain definitions such as the following:

```
\DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
```

This tells LaTeX that when `\rm` is used in the text it should switch to `\normalfont` and then select the roman family. When `\rm` is used in mathmode LaTeX will select the font that would be selected by `\mathrm`[1].

Build on standard classes

When upgrading your own document style you should consider to reimplement it by building on an existing Document Class. With the new features of LaTeX 2_ε this has become very easy. The advantage of this approach is that you don't have to maintain a whole lot of code that is probably basically a copy of the code in one of the standard document classes. (See below for a few examples of how to build your own document class on an existing class.) Some documentstyles written for LaTeX 2.09, such as `ltugboat`, contain a command such as `\input{article.sty}`. This was the only solution in LaTeX 2.09—to build a new documentstyle upon an existing style. But, there was no way of ensuring that the file `article.sty` which was found by LaTeX wasn't out of date. As you see in the examples below, it is now possible to ensure that you use a version of `article.cls` that was released after a certain date.

Suggested updates.

Apart from the essential changes to your document class, there are also a few changes that you are encouraged to make. Most of these changes have to do with the new possibilities the package and class writers interface gives you.

In a LaTeX 2.09 document style an option was declared by defining a command that starts with `\ds@` followed by the name of the option. Later on in the documentstyle the command `\@options` was called to execute the code for the options that were supplied by the user. For example, the document style `article` contained the following lines of code:

[1] See *LaTeX 2_ε font selection* for more details.

```
...
\def\ds@twoside{\@twosidetrue\@mparswitchtrue}
\def\ds@draft{\overfullrule 5\p@}
...
\@options
...
```

This code fragment defined two options, twoside and draft.

The same effect can be achieved by using LaTeX 2ε syntax, as is shown by the following code fragment from the document class `article`:

```
...
\DeclareOption{oneside}{\@twosidefalse \@mparswitchfalse}
\DeclareOption{twoside}{\@twosidetrue  \@mparswitchtrue}
\DeclareOption{draft}{\setlength\overfullrule{5pt}}
\DeclareOption{final}{\setlength\overfullrule{0pt}}
...
\ProcessOptions
```

As you can see, the intention of this code is easier to understand.

I consider it good practice, when writing packages and classes, to use the higher level LaTeX commands as much as possible. So instead of using `\def...` I recommend using one of `\newcommand`, `\renewcommand` or `\providecommand`. This makes it less likely that you inadvertently redefine a command, giving unexpected results.

When you define an environment use the commands `\newenvironment` or `\renewenvironment` instead of `\def\foo{...}` and `\def\endfoo{...}`.

If you need to set or change the value of a ⟨*dimen*⟩ or ⟨*skip*⟩ register, use `\setlength`.

The advantage of this practice is that your code is more readable and that it is less likely to break when future versions of LaTeX are made available.

Some packages and document styles had to redefine the `\begin{document}` or `\end{document}` commands to achieve their goal. This is no longer necessary. The "hooks" `\AtBeginDocument` and `\AtEndDocument` are now available. They make it more likely that your package will work together with someone else's.

When a document class needs to pass information to the user, you can use one of the commands `\ClassInfo`, `\ClassWarning`,
`\ClassWarningNoLine` or `\ClassError`. A similar set of commands exists for packages.

Be colour safe

One of the new features of LaTeX 2ε is the support for coloured documents. To create a document that contains colour you need:

- the `color` package, which is part of the LaTeX 2ε distribution;
- a driver which supports colour—dvips by Tomas Rokicki is an example of such a driver;
- colour safe macros.

The first two points are probably obvious, the third point needs some explanation. TeX has no knowledge of colour, therefore the macros need to keep track of the colour. To achieve that, various changes have been made to the kernel of LaTeX. This has been done in such a way that the changes are 'dormant' when the `color` package isn't used. As an example, here is the current definition[2] of the LaTeX command `\sbox`:

[2]Shown here only as an illustration; the actual implementation may change.

```
\def\sbox#1#2{\setbox#1\hbox{\color@@setgroup#2\color@@endgroup}}
```

The extra level of grouping is activated by the `color` package and is needed to keep colour changes local. For more information about being 'color safe' you should read the documentation that comes with the `color` package. If you use the LaTeX commands for boxing sunch as `\mbox`, `\sbox`, `\fbox`, etc. instead of the low level commands `\hbox`, `\vbox` and `\setbox`, your code will be automatically 'colour safe'.

6.3 Upgrading existing 'styles'—an example tour

6.3.1 A minimal class

Most of the work of a class or package is in defining new commands, or changing the appearance of documents. This is done in the body of the class or package, using commands such as `\newcommand`, `\setlength` and `\sbox` (or `\savebox`).
However, there are some new commands for helping class and package writers. These are described in detail in *LaTeX 2ε for class and package writers*.
There are three definitions that every class *must* provide. These are `\normalsize`, `\textwidth` and `\textheight`. So a minimal document class file is:

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{minimal}[1994/12/01 Minimal class]
\renewcommand{\normalsize}{\fontsize{10}{12}\selectfont}
\setlength{\textwidth}{6.5in}
\setlength{\textheight}{8in}
```

However, most classes will provide more than this!

6.3.2 Extending a class with new commands

The first example shows how you can extend an existing class with a few extra commands. Suppose you call your new class `extart`. It could start off with the following code:

```
%------------ Identification ------------%
\NeedsTeXFormat{LaTeX2e}[1994/12/01]
\ProvidesClass{extart}[1994/08/01 v2.0j Article like class with new commands]
```

This first line tells LaTeX that your code was written for LaTeX 2ε, released after june first, 1994. The second line informs LaTeX that this file provides the document class `extart`, dated august 1, 1994, and with version 2.0j.

```
%------------ Option handling -----------%
\DeclareOption*{\PassOptionsToClass{\CurrentOption}{article}}
```

The code above instructs LaTeX to pass on every option the user asked for to the document class `article`.

```
\ProcessOptions
%------------ Load other class ----------%
\LoadClass[a4paper]{article}[1994/12/01]
```

The command `\ProcessOptions` executes the code associated with each option the user specified. The `\LoadClass` command subsequently loads the class file. The first optional argument to `\LoadClass` passes the option `a4paper` to the class; the second optional argument to `\LoadClass` asks for `article.cls` dated june first, 1994, or later. Note that if you change your mind and load `report` instead you also have to change the second argument of `\PassOptionsToClass`.

```
%------------ Extra command -------------%
\newcommand\foo{\typeout{Hello world!}}
...
```

The rest of the file contains the extra code you need such as the definition of the command `\foo`.

6.3.3 Changing the layout produced by another class

The first few lines of a class that modifies the layout of an existing class would look much the same as in the example above.

```
%------------ Identification ------------%
\NeedsTeXFormat{LaTeX2e}[1994/12/01]
\ProvidesClass{review}[1994/08/01 v1.0 Article like class with changed layout]
%------------ Option handling -----------%
\DeclareOption*{\PassOptionsToClass{\CurrentOption}{article}}
\ProcessOptions
%------------ Load other class ----------%
\LoadClass{article}[1994/12/01]
```

Suppose we have to print on paper 7 inch wide and 9.875 inch tall. The text should measure 5.5 inch by 8.25 inch

```
%------------- Layout of text -----------%
\setlength{\paperwidth}{7in}
\setlength{\paperheight}{9.875in}
\setlength{\textwidth}{5.5in}
\setlength{\textheight}{8.25in}
```

What we have to do now is position the body of the text in a proper place on the paper.

```
\setlength{\topmargin}{-.5625in}
\setlength{\oddsidemargin}{-.25in}
\setlength{\evensidemargin}{-.25in}
\setlength{\marginparwidth}{.25in}
\setlength{\headsep}{.1875in}
```

We could go on and modify other aspects of the design of the text, but that is beyond the scope of this article.

6.3.4 Extending a class with new options

As before, we start the document class with some identification.

```
%------------ Identification ------------%
\NeedsTeXFormat{LaTeX2e}[1994/12/01]
\ProvidesClass{optart}[1994/08/01 v1.0 Article like class with extra options]
```

Suppose you want to be able to print a document in 9pt type, or when you want to be loud, print it in 14pt type. You know that the standard LaTeX classes contain the command
`\input{size1\@ptsize.clo}`
just after the execution of `\ProcessOptions`. Supposing you don't want to print an article in 19pt type, you can use the file name `size9.clo` to implement your design for a layout that assumes the type size is 9pt. To implement a design for 14pt type you create the file `size14.clo`.

Adding the options to your extended document class is done by the following two lines of code:

```
\DeclareOption{9pt}{\renewcommand\@ptsize{9}}
\DeclareOption{14pt}{\renewcommand\@ptsize{4}}
```

All other options have to be passed on to the `article` class.

```
%------------ Option handling -----------%
\DeclareOption*{%
  \PassOptionsToClass{\CurrentOption}
                     {article}}
\ProcessOptions
%------------ Load other class ----------%
\LoadClass{article}[1994/12/01]
```

6.3.5 A real life example

Apart from adding options to an existing document class it is also possible to *disable* options that are allowed by the document class you are building upon. An example of this is the document class `ltxdoc`, used by the LaTeX3 project team for the documented source code of LaTeX. It contains the following lines of code:

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{ltxdoc}[1994/05/27 v2.0n Standard LaTeX documentation class]
\DeclareOption{a5paper}{\@latexerr{Option not supported}{}}
\DeclareOption*{\PassOptionsToClass  {\CurrentOption}{article}}
...
```

The interesting bit is the line that associates the option `a5paper` with an error message. When someone specifies the `a5paper` option to the class `ltxdoc` he will be warned that this document class does not support printing on A5 paper.

This document class allows customization by checking if a file `ltxdoc.cfg` exists. If a file with that name is found the user is told that the file is read in.

```
\InputIfFileExists{ltxdoc.cfg}
{\typeout{%
 *************************************^^J%
 * Local config file ltxdoc.cfg used^^J%
 *************************************}}
{}
```

Such a configuration file might contain the instruction to use A4 paper for printing:
`\PassOptionsToClass{a4paper}{article}`
When the configuration file is read, the options are processed and the `article` class is loaded.

```
\ProcessOptions
\LoadClass{article}
```

Then the package `doc` is required. This package is needed to print documented TeX source code, which the document class `ltxdoc` is made for.

```
\RequirePackage{doc}
```

The last line from this document class that is interesting is the following:

```
\AtBeginDocument{\MakeShortVerb{\|}}
```

This instructs LaTeX to store the command
`\MakeShortVerb` together with its argument (`\|`) to be executed when `\begin{document}` is encountered.

6.3.6 Compatibility with LaTeX 2.09

Upwards compatibility is provided by the compatibility mode of LaTeX 2ε. This mode was introduced to be able to run old LaTeX 2.09 documents through LaTeX 2ε, yielding (almost) the same result. If this is what you need to achieve, than you may be pleased to know that the `\if@compatibility` switch can be used to test for compatibility mode. Using this switch, you can develop a full blown LaTeX 2ε Package or Document Class out of a LaTeX 2.09 style file and yet still be able to print your old documents without changing them.

6.3.7 Possible Pitfalls while upgrading

Some mistakes that might be easily made and that can lead to unexpected results:

- You declare options in your package using
  `\DeclareOption` but forget to call
  `\ProcessOptions`. LaTeX will give an error, 'unprocessed options' unless sometimes other errors in the class file intervened and prevent the system detecting this mistake.

- The usage of either `\footheight`, `\@maxsep` or `\@dblmaxsep` outside of compatibility mode will lead to a complaint from TeX about an unknown command sequence.

- With LaTeX 2.09 the order in which options to a documentstyle were specified was *very* significant. A document would fail if the options were given in the wrong order. By default LaTeX 2ε does *not* process the options in the order that they were specified in the document. It rather processes them in the order that they are declared in the class or package file. When the order of processing the options is relevant to your code you can use the command `\ProcessOptions*`. This will make LaTeX 2ε evaluate the options in the order that they were specified in by the user.

  For the babel package for instance, the order of processing the options is significant. The last language specified in the option list will be the one the document starts off with.

Bibliography

[1] Leslie Lamport. *LaTeX—A Document Preparation System. Second edition* Addison-Wesley, Reading, USA, 1994.

[2] Michel Goossens, Frank Mittelbach and Alexander Samarin. *The LaTeX Companion*. Addison-Wesley, Reading, USA, 1994.

[3] Frank Mittelbach and Rainer Schöpf. Towards LaTeX 2.10. *TUGBoat*, 10(3):400–401, November 1990.

[4] Frank Mittelbach. E-TeX: Guidelines for future TeX extensions. *TUGBoat*, 11(3):337–345, September 1990.

[5] Frank Mittelbach and Chris Rowley. LaTeX 2.09 $\hookrightarrow$ LaTeX3. *TUGBoat*, 13(1):96-101, April 1992.

[6] Chris Rowley. LaTeX3 update. *TUGBoat*, 13(3):390-391, October 1992.

[7] Frank Mittelbach and Chris Rowley and Michael Downes. Volunteer work for the LaTeX3 project. *TUGBoat*, 13(4):510-515, December 1992.

[8] Frank Mittelbach and Chris Rowley. Volunteer work for the LaTeX3 project. *TeX and TUG NEWS*, 3(1):7-11, January 1994.

[9] *AMSFonts version 2.1—user's guide*, Amer. Math. Soc., Providence, RI, 1994; distributed with the AMSFonts package.

[10] *Instructions for preparation of papers and monographs—AMS-LaTeX*, Amer. Math. Soc., Providence, RI, 1994.

[11] Donald E. Knuth. *The TeXbook*, Addison-Wesley, Reading, 1990.

[12] D.E. Knuth. *Computers and Typesetting (Volumes A-E)*. Reading, 1986.

[13] Michael Spivak. *The joy of TeX*, 2nd revised ed., Amer. Math. Soc., Providence, RI, 1990.

[14] Ph.Defert and I.Reguero. *ASIS, The Installer's Guide* CN-CO-152. CERN, 1992.

[15] Michel Goossens and Eric van Herwijnen. Scientific Text Processing *International Journal of Modern Physics C*, Vol. 3, No 3 (1992) 479-546.

[16] Michel Goossens and Eric van Herwijnen. The Elementatry Particle Entity Notation (PEN) Scheme *TUGboat*, 13,2:201–207, 1992.

[17] Frank Mittelbach and Rainer Schöpf. The New Font Selection - User Interface to Standard LaTeX. *TUGboat*, 11,2:297–305, 1990.

[18] Hugh Williamson. *Methods for Book Design, Third Edition*. Yale University Press, Newhaven, London, 1985.

[19] Frank Mittelbach and Rainer Schöpf. A New Font Selection Scheme for TeX Macro Packages. *TUGboat*, 10,2:222–238, 1989.

[20] Karl Berry. Filenames for Fonts. *TUGboat*, 11:517–520, 1990.

[21] D.E. Knuth. Virtual Fonts: More Fun for Grand Wizards. *TUGboat*, 11:13–23, 1990.

[22] Adobe Systems Incorporated. *PostScript Language Reference Manual*. Addison-Wesley, Reading, Second edition, 1990.

[23] M.D. Spivak. *The Joy of TeX, 2nd Edition*. American Mathematical Society, Providence, Rhode Island, 1990.

[24] The LaTeX3 Project team. *LaTeX 2ε for Authors.* A document provided in the LaTeX 2ε distribution in file `usrguide.tex`

[25] The LaTeX3 Project team. *LaTeX 2ε for class and package writers.* A document provided in the LaTeX 2ε distribution in file `clsguide.tex`

[26] The LaTeX3 Project team. *LaTeX 2ε font selection.* A document provided in the LaTeX 2ε distribution in file `fntguide.tex`

[27] International Union of pure and applied Physics. *Symbols, Units, Nomenclature and fundamental Constants in Physics*. Physica, 146A:1–67, 1987.

[28] D.E. Lowe. *A Guide to international recommendations on names and symbols for quantities and on units of measurements*. World Health Organization, Geneva, 1975.

[29] Particle Data Group. *Review of particle properties.* Physics Review D, 50, Part 1, pages 1173-1826, August 1994.

[30] E. van Herwijnen. *Practical SGML.* Wolters-Kluwer Academic Publishers, Boston, 1990.

[31] American National Standards Institute. *American National Standard for Electronic Manuscript Preparation and Markup* ANSI/NISO Z39.59-1988, 1988.

[32] Particle Data Group. *A Guide to Experimental Elementary Particle Physics Literature (1985-1989).* Lawrence Berkeley Laboratory, LBL-90 Revised, UC-414, November 1990.

Appendix A: LaTeX tabular overview

Tables A.1 to A.15 show all characters available in standard LaTeX and some LaTeX commands, while Table A.16 shows the supplementary $\mathcal{AMS}$ symbols which become available when loading the `amsfonts` package.

α	`\alpha`	β	`\beta`	γ	`\gamma`	δ	`\delta`	ϵ	`\epsilon`
ε	`\varepsilon`	ζ	`\zeta`	η	`\eta`	θ	`\theta`	ϑ	`\vartheta`
ι	`\iota`	κ	`\kappa`	λ	`\lambda`	μ	`\mu`	ν	`\nu`
ξ	`\xi`	o	`o`	π	`\pi`	ϖ	`\varpi`	ρ	`\rho`
ϱ	`\varrho`	σ	`\sigma`	ς	`\varsigma`	τ	`\tau`	υ	`\upsilon`
ϕ	`\phi`	φ	`\varphi`	χ	`\chi`	ψ	`\psi`	ω	`\omega`
Γ	`\Gamma`	Δ	`\Delta`	Θ	`\Theta`	Λ	`\Lambda`	Ξ	`\Xi`
Π	`\Pi`	Σ	`\Sigma`	Υ	`\Upsilon`	Φ	`\Phi`	Ψ	`\Psi`
Ω	`\Omega`								

Table A.1: Greek Letters

$\pm$	`\pm`	$\cap$	`\cap`	$\diamond$	`\diamond`	$\oplus$	`\oplus`
$\mp$	`\mp`	$\cup$	`\cup`	$\bigtriangleup$	`\bigtriangleup`	$\ominus$	`\ominus`
$\times$	`\times`	$\uplus$	`\uplus`	$\bigtriangledown$	`\bigtriangledown`	$\otimes$	`\otimes`
$\div$	`\div`	$\sqcap$	`\sqcap`	$\triangleleft$	`\triangleleft`	$\oslash$	`\oslash`
$\ast$	`\ast`	$\sqcup$	`\sqcup`	$\triangleright$	`\triangleright`	$\odot$	`\odot`
$\star$	`\star`	$\vee$	`\vee`	$\lhd$	`\lhd`	$\bigcirc$	`\bigcirc`
$\circ$	`\circ`	$\wedge$	`\wedge`	$\rhd$	`\rhd`	$\dagger$	`\dagger`
$\bullet$	`\bullet`	$\setminus$	`\setminus`	$\unlhd$	`\unlhd`	$\ddagger$	`\ddagger`
$\cdot$	`\cdot`	$\wr$	`\wr`	$\unrhd$	`\unrhd`	$\amalg$	`\amalg`
$+$	`+`	$-$	`-`				

Table A.2: Binary Operation Symbols

$\leq$	`\leq`	$\geq$	`\geq`	$\equiv$	`\equiv`	$\models$	`\models`	$\prec$	`\prec`
$\succ$	`\succ`	$\sim$	`\sim`	$\perp$	`\perp`	$\preceq$	`\preceq`	$\succeq$	`\succeq`
$\simeq$	`\simeq`	$\mid$	`\mid`	$\ll$	`\ll`	$\gg$	`\gg`	$\asymp$	`\asymp`
$\parallel$	`\parallel`	$\subset$	`\subset`	$\supset$	`\supset`	$\approx$	`\approx`	$\bowtie$	`\bowtie`
$\subseteq$	`\subseteq`	$\supseteq$	`\supseteq`	$\cong$	`\cong`	$\Join$	`\Join`	$\sqsubset$	`\sqsubset`
$\sqsupset$	`\sqsupset`	$\neq$	`\neq`	$\smile$	`\smile`	$\sqsubseteq$	`\sqsubseteq`	$\sqsupseteq$	`\sqsupseteq`
$\doteq$	`\doteq`	$\frown$	`\frown`	$\in$	`\in`	$\ni$	`\ni`	$\propto$	`\propto`
$=$	`=`	$\vdash$	`\vdash`	$\dashv$	`\dashv`	$<$	`<`	$>$	`>`

Table A.3: Relation Symbols

$,$	`,`	$;$	`;`	$:$	`\colon`	$\ldotp$	`\ldotp`	$\cdotp$	`\cdotp`

Table A.4: Punctuation Symbols

$\leftarrow$	`\leftarrow`	$\longleftarrow$	`\longleftarrow`	$\uparrow$	`\uparrow`
$\Leftarrow$	`\Leftarrow`	$\Longleftarrow$	`\Longleftarrow`	$\Uparrow$	`\Uparrow`
$\rightarrow$	`\rightarrow`	$\longrightarrow$	`\longrightarrow`	$\downarrow$	`\downarrow`
$\Rightarrow$	`\Rightarrow`	$\Longrightarrow$	`\Longrightarrow`	$\Downarrow$	`\Downarrow`
$\leftrightarrow$	`\leftrightarrow`	$\longleftrightarrow$	`\longleftrightarrow`	$\updownarrow$	`\updownarrow`
$\Leftrightarrow$	`\Leftrightarrow`	$\Longleftrightarrow$	`\Longleftrightarrow`	$\Updownarrow$	`\Updownarrow`
$\mapsto$	`\mapsto`	$\longmapsto$	`\longmapsto`	$\nearrow$	`\nearrow`
$\hookleftarrow$	`\hookleftarrow`	$\hookrightarrow$	`\hookrightarrow`	$\searrow$	`\searrow`
$\leftharpoonup$	`\leftharpoonup`	$\rightharpoonup$	`\rightharpoonup`	$\swarrow$	`\swarrow`
$\leftharpoondown$	`\leftharpoondown`	$\rightharpoondown$	`\rightharpoondown`	$\nwarrow$	`\nwarrow`

Table A.5: Arrow Symbols

$\ldots$	`\ldots`	$\cdots$	`\cdots`	$\vdots$	`\vdots`	$\ddots$	`\ddots`	$\aleph$	`\aleph`
$\prime$	`\prime`	$\forall$	`\forall`	∞	`\infty`	$\hbar$	`\hbar`	$\emptyset$	`\emptyset`
$\exists$	`\exists`	∇	`\nabla`	$\surd$	`\surd`	$\Box$	`\Box`	$\triangle$	`\triangle`
$\Diamond$	`\Diamond`	$\imath$	`\imath`	$\jmath$	`\jmath`	ℓ	`\ell`	$\neg$	`\neg`
$\top$	`\top`	$\flat$	`\flat`	$\natural$	`\natural`	$\sharp$	`\sharp`	$\wp$	`\wp`
$\bot$	`\bot`	$\clubsuit$	`\clubsuit`	$\diamondsuit$	`\diamondsuit`	$\heartsuit$	`\heartsuit`	$\spadesuit$	`\spadesuit`
$\backslash$	`\backslash`	$\Re$	`\Re`	$\Im$	`\Im`	$\angle$	`\angle`	∂	`\partial`
$\|$	`|`	$\mho$	`\mho`	$.$	`.`	$\vert$	`|`		

Table A.6: Miscellaneous Symbols

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

Table A.7: Variable-sized Symbols

`\arccos`	`\cos`	`\csc`	`\exp`	`\ker`	`\limsup`	`\min`	`\sinh`
`\arcsin`	`\cosh`	`\deg`	`\gcd`	`\lg`	`\ln`	`\Pr`	`\sup`
`\arctan`	`\cot`	`\det`	`\hom`	`\lim`	`\log`	`\sec`	`\tan`
`\arg`	`\coth`	`\dim`	`\inf`	`\liminf`	`\max`	`\sin`	`\tanh`

Table A.8: Log-like Symbols

$($	`(`	$)$	`)`	$\uparrow$	`\uparrow`	$\Uparrow$	`\Uparrow`
$[$	`[`	$]$	`]`	$\downarrow$	`\downarrow`	$\Downarrow$	`\Downarrow`
$\{$	`\{`	$\}$	`\}`	$\updownarrow$	`\updownarrow`	$\Updownarrow$	`\Updownarrow`
$\lfloor$	`\lfloor`	$\rfloor$	`\rfloor`	$\lceil$	`\lceil`	$\rceil$	`\rceil`
$\langle$	`\langle`	$\rangle$	`\rangle`	$/$	`/`	$\backslash$	`\backslash`
$\vert$	`|`	$\|$	`|`				

Table A.9: Delimiters

$\rmoustache$	\rmoustache	$\lmoustache$	\lmoustache	$\rgroup$	\rgroup	$\lgroup$	\lgroup
$\arrowvert$	\arrowvert	$\Arrowvert$	\Arrowvert	$\bracevert$	\bracevert		

Table A.10: Large Delimiters

$\hat{a}$	\hat{a}	$\acute{a}$	\acute{a}	$\bar{a}$	\bar{a}	$\dot{a}$	\dot{a}	$\breve{a}$	\breve{a}
$\check{a}$	\check{a}	$\grave{a}$	\grave{a}	$\vec{a}$	\vec{a}	$\ddot{a}$	\ddot{a}	$\tilde{a}$	\tilde{a}

Table A.11: Math mode accents

ò	\`o	ó	\'o	ô	\^o	õ	\~o	ō	\=o	ȯ	\.o		
ŏ	\u{o}	ǒ	\v{o}	ő	\H{o}	O̧	\c{O}	ọ	\d{o}	o̲	\b{o}	o͡o	\t{oo}
œ	{\oe}	Œ	{\OE}	æ	{\ae}	Æ	{\AE}	å	{\aa}	Å	{\AA}		
ø	{\o}	Ø	{\O}	ł	{\l}	Ł	{\L}	¿	?`	¡	!`		

Table A.12: Accents and foreign language symbols in text mode

\tiny	Some text	\scriptsize	Some text
\footnotesize	Some text	\small	Some text
\normalsize	Some text	\large	Some text
\Large	Some text	\LARGE	Some text
\huge	Some text	\Huge	Some text

Table A.13: Font sizes — text mode only

\rmfamily	Roman	\itshape	*Italic*	\scshape	SMALL CAPS	\bfseries	**Bold Face**
\slshape	*Slanted*	\sffamily	Sans Serif	\ttfamily	`Typewriter`		

Table A.14: Font types

$\widetilde{abc}$	\widetilde{abc}	$\widehat{abc}$	\widehat{abc}	$\overleftarrow{abc}$	\overleftarrow{abc}
$\overrightarrow{abc}$	\overrightarrow{abc}	$\overline{abc}$	\overline{abc}	$\underline{abc}$	\underline{abc}
$\overbrace{abc}$	\overbrace{abc}	$\underbrace{abc}$	\underbrace{abc}	$\sqrt{abc}$	\sqrt{abc}
$\sqrt[n]{abc}$	\sqrt[n]{abc}	f'	f'	$\frac{abc}{xyz}$	\frac{abc}{xyz}

Table A.15: Some other constructs

$\boxdot$	\boxdot	$\boxplus$	\boxplus	$\boxtimes$	\boxtimes	$\square$	\square
$\blacksquare$	\blacksquare	$\centerdot$	\centerdot	$\lozenge$	\lozenge	$\blacklozenge$	\blacklozenge
$\circlearrowright$	\circlearrowright	$\circlearrowleft$	\circlearrowleft	$\rightleftharpoons$	\rightleftharpoons	$\leftrightharpoons$	\leftrightharpoons
$\boxminus$	\boxminus	$\Vdash$	\Vdash	$\Vvdash$	\Vvdash	$\vDash$	\vDash
$\twoheadrightarrow$	\twoheadrightarrow	$\twoheadleftarrow$	\twoheadleftarrow	$\leftleftarrows$	\leftleftarrows	$\rightrightarrows$	\rightrightarrows
$\upuparrows$	\upuparrows	$\downdownarrows$	\downdownarrows	$\upharpoonright$	\upharpoonright	$\downharpoonright$	\downharpoonright
$\upharpoonleft$	\upharpoonleft	$\downharpoonleft$	\downharpoonleft	$\rightarrowtail$	\rightarrowtail	$\leftarrowtail$	\leftarrowtail
$\leftrightarrows$	\leftrightarrows	$\rightleftarrows$	\rightleftarrows	$\Lsh$	\Lsh	$\Rsh$	\Rsh
$\rightsquigarrow$	\rightsquigarrow	$\leftrightsquigarrow$	\leftrightsquigarrow	$\looparrowleft$	\looparrowleft	$\looparrowright$	\looparrowright
$\circeq$	\circeq	$\succsim$	\succsim	$\gtrsim$	\gtrsim	$\gtrapprox$	\gtrapprox
$\multimap$	\multimap	$\therefore$	\therefore	$\because$	\because	$\doteqdot$	\doteqdot
$\triangleq$	\triangleq	$\precsim$	\precsim	$\lesssim$	\lesssim	$\lessapprox$	\lessapprox
$\eqslantless$	\eqslantless	$\eqslantgtr$	\eqslantgtr	$\curlyeqprec$	\curlyeqprec	$\curlyeqsucc$	\curlyeqsucc
$\preccurlyeq$	\preccurlyeq	$\leqq$	\leqq	$\leqslant$	\leqslant	$\lessgtr$	\lessgtr
$\backprime$	\backprime	$\risingdotseq$	\risingdotseq	$\fallingdotseq$	\fallingdotseq	$\succcurlyeq$	\succcurlyeq
$\geqq$	\geqq	$\geqslant$	\geqslant	$\gtrless$	\gtrless	$\sqsubset$	\sqsubset
$\sqsupset$	\sqsupset	$\vartriangleright$	\vartriangleright	$\vartriangleleft$	\vartriangleleft	$\trianglerighteq$	\trianglerighteq
$\trianglelefteq$	\trianglelefteq	$\bigstar$	\bigstar	$\between$	\between	$\blacktriangledown$	\blacktriangledown
$\blacktriangleright$	\blacktriangleright	$\blacktriangleleft$	\blacktriangleleft	$\vartriangle$	\vartriangle	$\blacktriangle$	\blacktriangle
$\triangledown$	\triangledown	$\eqcirc$	\eqcirc	$\lesseqgtr$	\lesseqgtr	$\gtreqless$	\gtreqless
$\lesseqqgtr$	\lesseqqgtr	$\gtreqqless$	\gtreqqless	$\Rrightarrow$	\Rrightarrow	$\Lleftarrow$	\Lleftarrow
$\veebar$	\veebar	$\barwedge$	\barwedge	$\doublebarwedge$	\doublebarwedge	$\angle$	\angle
$\measuredangle$	\measuredangle	$\sphericalangle$	\sphericalangle	$\varpropto$	\varpropto	$\smallsmile$	\smallsmile
$\smallfrown$	\smallfrown	$\Subset$	\Subset	$\Supset$	\Supset	$\Cup$	\Cup
$\Cap$	\Cap	$\curlywedge$	\curlywedge	$\curlyvee$	\curlyvee	$\leftthreetimes$	\leftthreetimes
$\rightthreetimes$	\rightthreetimes	$\subseteqq$	\subseteqq	$\supseteqq$	\supseteqq	$\bumpeq$	\bumpeq
$\Bumpeq$	\Bumpeq	$\lll$	\lll	$\ggg$	\ggg	$\circledS$	\circledS
$\pitchfork$	\pitchfork	$\dotplus$	\dotplus	$\backsim$	\backsim	$\backsimeq$	\backsimeq
$\complement$	\complement	$\intercal$	\intercal	$\circledcirc$	\circledcirc	$\circledast$	\circledast
$\circleddash$	\circleddash	$\ulcorner$	\ulcorner	$\urcorner$	\urcorner	$\llcorner$	\llcorner
$\lrcorner$	\lrcorner	$\yen$	\yen	$\checkmark$	\checkmark	$\circledR$	\circledR
$\maltese$	\maltese	$\lvertneqq$	\lvertneqq	$\gvertneqq$	\gvertneqq	$\nleq$	\nleq
$\ngeq$	\ngeq	$\nless$	\nless	$\ngtr$	\ngtr	$\nprec$	\nprec
$\nsucc$	\nsucc	$\lneqq$	\lneqq	$\gneqq$	\gneqq	$\nleqslant$	\nleqslant
$\ngeqslant$	\ngeqslant	$\lneq$	\lneq	$\gneq$	\gneq	$\npreceq$	\npreceq
$\nsucceq$	\nsucceq	$\precnsim$	\precnsim	$\succnsim$	\succnsim	$\lnsim$	\lnsim
$\gnsim$	\gnsim	$\nleqq$	\nleqq	$\ngeqq$	\ngeqq	$\precneqq$	\precneqq
$\succneqq$	\succneqq	$\precnapprox$	\precnapprox	$\succnapprox$	\succnapprox	$\lnapprox$	\lnapprox
$\gnapprox$	\gnapprox	$\nsim$	\nsim	$\varkappa$	\varkappa	$\varsubsetneq$	\varsubsetneq
$\varsupsetneq$	\varsupsetneq	$\nsubseteqq$	\nsubseteqq	$\nsupseteqq$	\nsupseteqq	$\subsetneqq$	\subsetneqq
$\supsetneqq$	\supsetneqq	$\varsubsetneqq$	\varsubsetneqq	$\varsupsetneqq$	\varsupsetneqq	$\subsetneq$	\subsetneq
$\supsetneq$	\supsetneq	$\nsubseteq$	\nsubseteq	$\nsupseteq$	\nsupseteq	$\nparallel$	\nparallel
$\nmid$	\nmid	$\nshortmid$	\nshortmid	$\nshortparallel$	\nshortparallel	$\nvdash$	\nvdash
$\nVdash$	\nVdash	$\nvDash$	\nvDash	$\nVDash$	\nVDash	$\ntrianglerighteq$	\ntrianglerighteq
$\ntrianglelefteq$	\ntrianglelefteq	$\ntriangleleft$	\ntriangleleft	$\ntriangleright$	\ntriangleright	$\nleftarrow$	\nleftarrow
$\nrightarrow$	\nrightarrow	$\nLeftarrow$	\nLeftarrow	$\nRightarrow$	\nRightarrow	$\nLeftrightarrow$	\nLeftrightarrow
$\nleftrightarrow$	\nleftrightarrow	$\divideontimes$	\divideontimes	$\varnothing$	\varnothing	$\nexists$	\nexists
$\mho$	\mho	$\backepsilon$	\backepsilon	$\beth$	\beth	$\gimel$	\gimel
$\daleth$	\daleth	$\lessdot$	\lessdot	$\gtrdot$	\gtrdot	$\ltimes$	\ltimes
$\rtimes$	\rtimes	$\shortmid$	\shortmid	$\shortparallel$	\shortparallel	$\smallsetminus$	\smallsetminus
$\thicksim$	\thicksim	$\thickapprox$	\thickapprox	$\approxeq$	\approxeq	$\succapprox$	\succapprox
$\precapprox$	\precapprox	$\curvearrowleft$	\curvearrowleft	$\curvearrowright$	\curvearrowright	$\digamma$	\digamma
$\hslash$	\hslash	$\hbar$	\hbar				

Table A.16: $\mathcal{AMS}$ symbols available with the amsfonts package

Appendix B: Elementary Particle Naming Scheme

This appendix discusses the Elementary Particle Entity Notation (PEN) scheme and its use with TEX and SGML. The scheme not only assures the typographic correctness of the printed symbols, but also eases the automatic extraction of information about the article by the recognition of the entity names.

B.1 Principles of the Particle Entity Notation (PEN)

The PEN naming scheme uses a notation which takes the following constraints into consideration:

- The notation should be able to describe all particles in the particle data summary tables from the "Review of Particle Properties" [29] and any future extension to these.
- The names should not exceed eight characters. This is the maximum length for entities in the SGML reference concrete syntax [30]. Staying within this limit means that the notation can be used with most SGML applications.
- Common particles such as protons and electrons should have short and simple names.
- Items that are indicated by superscripts are indicated before items that are indicated by subscripts.

Due to the eight character limitation the mass could not be added to the name. This means that in general an entity on its own is not adequate to unambiguously identify a particle, c.f. $\eta(549)$ and $\eta(1300)$ are both referred to as `Pgh`. Including mass dependences into the names is not a good idea anyway, since the mass can change with time when more precise measurements become available. The ambiguity was solved by adding a letter to the end of the name where a mass appears in the name in the particle data summary tables. Thus $\eta(549)$ is referred to as `Pgh` while $\eta(1300)$ is referred to as `Pgha`. Higher letters correspond to higher masses, in the order given in the tables.

The PEN scheme is independent of any text processing system. It has been implemented in TEX (in such a way that it may be used in all macro packages, e.g. LATEX) and SGML. The TEX implementation will print particle masses, which will be regularly updated according to the Review of Particle Properties publication. It is constructed so that the PEN name can be used both in mathematics and text mode.

In more detail, a PEN particle name is built from the following characters (going from left to right):

1. Start the entity with a recognized string (in the following this was chosen as `uppercase P`). This is necessary to uniquely identify entities as following the PEN convention.
2. The following letters act as an escape to signal a special interpretation of the string. Present escape sequences are:
    - `a` for anti particle (normally visually represented with a bar over the particle's name)
    - `b` for bottom particle
    - `c` for charmed particle
    - `g` for indicating the subsequent letter is Greek. The correspondence between Latin and Greek letters is based on the notation for mathematical Greek characters used by the AAP mathematical formula application [31]:

      ```
      <!NOTATION greek2 PUBLIC "+//ISBN 1-880124::NISO//NOTATION GREEK-2//EN">
      ```

      This one-letter correspondence is as follows:

Greek	name	code	Greek	name	code	Greek	name	code	Greek	name	code
α	alpha	`a`	A	Alpha	`A`	β	beta	`b`	B	Beta	`B`
γ	gamma	`g`	Γ	Gamma	`G`	δ	delta	`d`	Δ	Delta	`D`
ϵ	epsilon	`e`	E	Epsilon	`E`	ζ	zeta	`z`	Z	Zeta	`Z`
η	eta	`h`	H	Eta	`H`	θ	theta	`q`	Θ	Theta	`Q`
ι	iota	`i`	I	Iota	`I`	κ	kappa	`k`	K	Kappa	`K`
λ	lambda	`l`	Λ	Lambda	`L`	μ	mu	`m`	M	Mu	`M`
ν	nu	`n`	N	Nu	`N`	ξ	xi	`x`	Ξ	Xi	`X`
o	omicron	`o`	O	Omicron	`O`	π	pi	`p`	Π	Pi	`P`
ρ	rho	`r`	R	Rho	`R`	σ	sigma	`s`	Σ	Sigma	`S`
τ	tau	`t`	T	Tau	`T`	υ	upsilon	`u`	Υ	Upsilon	`U`
ϕ	phi	`f`	Φ	Phi	`F`	χ	chi	`c`	X	Chi	`C`
ψ	psi	`y`	Ψ	Psi	`Y`	ω	omega	`w`	Ω	Omega	`W`

- `q` for quark particle
- `s` for strange particle
- `S` for supersymmetric particle
- `t` for top particle

3. The one-letter name of the particle
4. Optionally followed by other information
   - `z` for zero, `i` for one, `ii` for two, `iii` for three, `iv` for four
   - `m` for minus, `p` for plus, `pm` for plus/minus
   - `pr` for prime
   - `st` for asterisk (star)
   - `L` for left handed, `R` for right handed
   - any one-letter particle name

B.2 Particle encodings according to the PEN Scheme

In table B.1 we show how to encode the particles from the summary tables of particle properties in the "Review of Particle Properties" [29] using the PEN convention. In the rightmost column we give the computer name of the particle, as defined by "A guide to Experimental Elementary Particle Physics Literature (1985-1989)" [32]. This is the name to be used when searching the Particle Data group's databases. Notice that these names cannot be used either for TeX or SGML, as they do not satisfy the constraints of the PEN scheme as defined above. When a name is marked as "`not available`", sometimes a charged or neutral version exists (not given in the table).

The TeX implementation is available as a package `pennames.sty` which should be input with a `\usepackage` command in LaTeX. To obtain a given symbol prefix the PEN name by a backslash ("\").

The SGML implementation exists as a public entity set, that can be included in SGML documents with the following entity definition:

```
<!ENTITY % PEN PUBLIC
   "+//ISBN 92-9083-041-7::CERN//ENTITIES Particle Entity Names//EN">
```

Refer to a particle entity by prefixing its name by an ampersand ("&") and suffixing it with a semi-colon (";").

Table B.1: PEN names for elementary particles in PDG list

PEN	symbol	conventional name	computer name
Gauge and Higgs bosons			
Pgg	γ	gamma	GAMMA
PW	W	W boson	W
PWp	W^+	W plus	W+
PWm	W^-	W minus	W-
PZz	Z^0	Z zero	Z
PHz	H^0	Higgs zero	not available
PHpm	$H^\pm$	Higgs plus/minus	HIGGS+-
PWR	W_R	right-handed W	not available
PWpr	W'	W prime	WPRIME
PZLR	Z_{LR}	left-right handed Z	not available
PZgc	Z_χ	Z chi	not available
PZgy	Z_ψ	Z psi	not available
PZge	Z_η	Z eta	not available
PZi	Z_1	Z one	not available
PAz	A^0	axion	AXION
Leptons			
Pgne	ν_e	electron neutrino	NUE
Pagne	$\overline{\nu}_e$	anti electron neutrino	NUEBAR
Pgngm	ν_μ	muon neutrino	NUMU
Pagngm	$\overline{\nu}_\mu$	anti muon neutrino	NUMUBAR
Pgngt	ν_τ	tau neutrino	NUTAU
Pagngt	$\overline{\nu}_\tau$	anti tau neutrino	NUTAUBAR
Pe	e	electron	not available
Pep	e^+	positron	E+
Pem	e^-	e minus	E-
Pgm	μ	muon	not available
Pgmm	μ^-	mu minus	MU-
Pgmp	μ^+	mu plus	MU+
Pgt	τ	tau	not available
PLpm	$L^\pm$	charged lepton	LEPTON+-
PLz	L^0	stable neutral heavy lepton	not available
PEz	E^0	neutral para- or ortho-lepton	not available
Light Unflavored Mesons (S=C=B=0)			
Pgp	π	pion	PI
Pgpm	π^-	pi minus	PI-
Pgpp	π^+	pi plus	PI+
Pgppm	$\pi^\pm$	pi plus/minus	PI+-
Pgpz	π^0	pi zero	PI0
Pgh	η	eta	ETA
Pgr	$\rho(770)$	rho	RHO(770)
Pgo	$\omega(783)$	omega	OMEGA(783)
Pghpr	$\eta'(958)$	eta prime	ETAPRIME(958)

Table B.1: PEN names (*continued*)

PEN	symbol	conventional name	computer name
Pfz	$f_0(975)$	f zero	F0(975)
Paz	$a_0(980)$	a zero	A0(980)
Pgf	$\phi(1020)$	phi	PHI(1020)
Phia	$h_1(1170)$	h one	H1(1170)
Pbi	$b_1(1235)$	b one	not available
Pai	$a_1(1260)$	a one	A1(1260)
Pfii	$f_2(1270)$	f two	F2(1270)
Pfi	$f_1(1285)$	f one	F1(1285)
Pgha	$\eta(1295)$	eta 1295	ETA(1295)
Pgpa	$\pi(1300)$	pion 1300	not available
Paii	$a_2(1320)$	a two	A2(1320)
Pgoa	$\omega(1390)$	omega 1390	not available
Pfza	$f_0(1400)$	f zero 1400	F0(1400)
Pfia	$f_1(1390)$	f one 1420	F1(1420)
Pghb	$\eta(1440)$	eta 1440	ETA(1440)
Pgra	$\rho(1450)$	rho 1450	not available
Pfib	$f_1(1510)$	f one 1510	F1(1510)
Pfiipr	$f'_2(1525)$	f two prime	F2PRIME(1525)
Pfzb	$f_0(1590)$	f zero 1590	F0(1590)
Pgob	$\omega(1600)$	omega 1600	not available
Pgoiii	$\omega_3(1670)$	omega three	OMEGA3(1670)
Pgpii	$\pi_2(1670)$	pi two	PI2(1670)
Pgfa	$\phi(1680)$	phi 1680	PHI(1680)
Pgriii	$\rho_3(1690)$	rho three	not available
Pgrb	$\rho(1700)$	rho 1700	RHO(1700)
Pfiia	$f_2(1720)$	f two 1720	F2(1720)
Pgfiii	$\phi_3(1850)$	phi three	PHI3(1850)
Pfiib	$f_2(2010)$	f two 2010	F2(2010)
Pfiv	$f_4(2050)$	f four	F4(2050)
Pfiic	$f_2(2300)$	f two 2300	F2(2300)
Pfiid	$f_2(2340)$	f two 2340	F2(2340)
Strange Mesons (S=$\pm$1, C=B=0)			
PK	K	kaon	K
PKpm	$K^\pm$	K plus minus	K+-
PKp	K^+	K plus	K+
PKm	K^-	K minus	K-
PKz	K^0	K zero	K0
PaKz	$\overline{K}^0$	anti K-zero	KBAR0
PKgmiii	$K_{\mu 3}$	K mu three	not available
PKeiii	K_{e3}	K e three	not available
PKzS	K^0_S	K zero short	not available
PKzL	K^0_L	K zero long	not available
PKzgmiii	$K^0_{\mu 3}$	K zero mu three	not available
PKzeiii	K^0_{e3}	K zero e three	not available

Table B.1: PEN names (*continued*)

PEN	symbol	conventional name	computer name
PKst	$K^*(892)$	K star	not available
PKi	$K_1(1270)$	K one	K1(1270)
PKsta	$K^*(1370)$	K star (1370)	not available
PKia	$K_1(1400)$	K one (1400)	not available
PKstz	$K_0^*(1430)$	K star zero (1430)	not available
PKstii	$K_2^*(1430)$	K star two (1430)	not available
PKstb	$K^*(1680)$	K star (1680)	not available
PKii	$K_2(1770)$	K two (1770)	not available
PKstiii	$K_3^*(1780)$	K star three	not available
PKstiv	$K_4^*(2045)$	K star four	not available
Charmed Mesons (C=±1)			
PD	D	D	D
PaD	$\overline{D}$	anti D	DBAR
PDpm	$D^{\pm}$	D plus/minus	D+-
PDm	D^-	D minus	D-
PDp	D^+	D plus	D+
PDz	D^0	D zero	D0
PaDz	$\overline{D}^0$	anti D zero	DBAR0
PDstpm	$D^*(2010)^{\pm}$	D star plus/minus	D*(2010)+-
PDstz	$D^*(2010)^0$	D star zero	D*(2010)0
PDiz	$D_1(2420)^0$	D one zero	D1(2420)0
PDstiiz	$D_2^*(2460)^0$	D star two zero	D2*(2460)0
Charmed Strange Mesons (C=S=±1)			
PsDp	D_s^+	D s plus	D/S+
PsDm	D_s^-	D s minus	D/S-
PsDst	D_s^*	D s star	D/S*
PsDipm	$D_{s1}(2536)^{\pm}$	D s one plus/minus	not available
Bottom Mesons (B=±1)			
PB	B	B	B
PBp	B^+	B plus	B+
PBm	B^-	B minus	B-
PBpm	$B^{\pm}$	B plus/minus	B+-
PBz	B^0	B zero	B0
PaB	$\overline{B}$	anti B	BBAR
PaBz	$\overline{B}^0$	anti B zero	BBAR0
Pcgh	$\eta_c(1S)$	eta c	ETA/C(1S)
PJgy	$J/\psi(1S)$	J psi	J/PSI(1S)
Pcgcz	$\chi_{c0}(1P)$	chi c zero	CHI/C0(1P)
Pcgci	$\chi_{c1}(1P)$	chi c one	CHI/C1(1P)
Pcgcii	$\chi_{c2}(1P)$	chi c two	CHI/C2(1P)
Pgy	$\psi(2S)$	psi	PSI(2S)
Pgya	$\psi(3770)$	psi 3770	PSI(3770)
Pgyb	$\psi(4040)$	psi 4040	PSI(4040)

Table B.1: PEN names (*continued*)

PEN	symbol	conventional name	computer name
`Pgyc`	$\psi(4160)$	psi 4160	`PSI(4160)`
`Pgyd`	$\psi(4415)$	psi 4415	`PSI(4415)`
`PgU`	$\Upsilon(1S)$	Upsilon	`not available`
`Pbgcz`	$\chi_{b0}(1P)$	chi b zero	`CHI/B0(1P)`
`Pbgci`	$\chi_{b1}(1P)$	chi b one	`CHI/B1(1P)`
`Pbgcii`	$\chi_{b2}(1P)$	chi b two	`CHI/B2(1P)`
`PgUa`	$\Upsilon(2S)$	Upsilon (2S)	`UPSI(2S)`
`Pbgcza`	$\chi_{b0}(2P)$	chi b zero (2P)	`CHI/B0(2P)`
`Pbgcia`	$\chi_{b1}(2P)$	chi b one (2P)	`CHI/B1(2P)`
`Pbgciia`	$\chi_{b2}(2P)$	chi b two (2P)	`CHI/B2(2P)`
`PgUb`	$\Upsilon(3S)$	Upsilon (3S)	`UPSI(3S)`
`PgUc`	$\Upsilon(4S)$	Upsilon (4S)	`UPSI(4S)`
`PgUd`	$\Upsilon(10860)$	Upsilon (10860)	`UPSI(10860)`
`PgUe`	$\Upsilon(11020)$	Upsilon (11020)	`UPSI(11020)`
quarks			
`Pq`	q	quark	`QUARK`
`Paq`	$\overline{q}$	anti-quark	`QUARKBAR`
`Pqd`	q_d	down quark	`DQ`
`Paqd`	$\overline{q}_d$	anti down quark	`DQBAR`
`Pqu`	q_u	up quark	`UQ`
`Paqu`	$\overline{q}_u$	anti up quark	`UQBAR`
`Pqs`	q_s	strange quark	`SQ`
`Paqs`	$\overline{q}_s$	anti strange quark	`SQBAR`
`Pqc`	q_c	charmed quark	`CQ`
`Paqc`	$\overline{q}_c$	anti charmed quark	`CQBAR`
`Pqb`	q_b	bottom quark	`BQ`
`Paqb`	$\overline{q}_b$	anti bottom quark	`BQBAR`
`Pqt`	q_t	top quark	`TQ`
`Paqt`	$\overline{q}_t$	anti top quark	`TQBAR`
N Baryons (S=0, I=1/2)			
`Pp`	p	proton	`P`
`Pap`	$\overline{p}$	anti-proton	`PBAR`
`Pn`	n	neutron	`N`
`PNa`	$N(1440)P_{11}$	N (1440) P 11	`N(1440P11)`
`PNb`	$N(1520)D_{13}$	N (1520) D 13	`not available`
`PNc`	$N(1535)S_{11}$	N (1535) S 11	`not available`
`PNd`	$N(1650)S_{11}$	N (1650) S 11	`not available`
`PNe`	$N(1675)D_{15}$	N (1675) D 15	`not available`
`PNf`	$N(1680)F_{15}$	N (1680) F 15	`not available`
`PNg`	$N(1700)D_{13}$	N (1700) D 13	`not available`
`PNh`	$N(1710)P_{11}$	N (1710) P 11	`not available`
`PNi`	$N(1720)P_{13}$	N (1720) P 13	`not available`
`PNj`	$N(2190)G_{17}$	N (2190) G 17	`not available`
`PNk`	$N(2220)H_{19}$	N (2220) H 19	`not available`

Table B.1: PEN names (*continued*)

PEN	symbol	conventional name	computer name
PNl	$N(2250)G_{19}$	N (2250) G 19	not available
PNm	$N(2600)I_{1,11}$	N (2600) I 1,11	not available
Δ Baryons (S=0, I=3/2)			
PgDa	$\Delta(1232)P_{33}$	Delta (1232) P 33	DELTA(1232P33)
PgDb	$\Delta(1620)S_{31}$	Delta (1620) S 31	not available
PgDc	$\Delta(1700)D_{33}$	Delta (1700) D 33	not available
PgDd	$\Delta(1900)S_{31}$	Delta (1900) S 31	not available
PgDe	$\Delta(1905)F_{35}$	Delta (1905) F 35	not available
PgDf	$\Delta(1910)P_{31}$	Delta (1910) P 31	not available
PgDh	$\Delta(1920)P_{33}$	Delta (1920) P 33	not available
PgDi	$\Delta(1930)D_{35}$	Delta (1930) D 35	not available
PgDj	$\Delta(1950)F_{37}$	Delta (1950) F 37	not available
PgDk	$\Delta(2420)H_{3,11}$	Delta (2420) H 3,11	not available
Λ Baryons (S=−1, I=0)			
PgL	Λ	Lambda	LAMBDA
PagL	$\overline{\Lambda}$	anti Lambda	LAMBDABAR
PgLa	$\Lambda(1405)S_{01}$	Lambda (1405) S 01	LAMBDA(1405S01)
PgLb	$\Lambda(1520)D_{03}$	Lambda (1520) D 03	LAMBDA(1520D03)
PgLc	$\Lambda(1600)P_{01}$	Lambda (1600) P 01	not available
PgLd	$\Lambda(1670)S_{01}$	Lambda (1670) S 01	not available
PgLe	$\Lambda(1690)D_{03}$	Lambda (1690) D 03	not available
PgLf	$\Lambda(1800)S_{01}$	Lambda (1800) S 01	not available
PgLg	$\Lambda(1810)P_{01}$	Lambda (1810) P 01	not available
PgLh	$\Lambda(1820)F_{05}$	Lambda (1820) F 05	not available
PgLi	$\Lambda(1830)D_{05}$	Lambda (1830) D 05	not available
PgLj	$\Lambda(1890)P_{03}$	Lambda (1890) P 03	not available
PgLk	$\Lambda(2100)G_{07}$	Lambda (2100) G 07	not available
PgLl	$\Lambda(2110)F_{05}$	Lambda (2110) F 05	not available
PgLm	$\Lambda(2350)H_{09}$	Lambda (2350) H 09	not available
Σ Baryons (S=−1, I=1)			
PgSm	Σ^-	Sigma minus	SIGMA-
PgSp	Σ^+	Sigma plus	SIGMA+
PagSm	$\overline{\Sigma}^-$	anti Sigma minus	SIGMABAR-
PagSp	$\overline{\Sigma}^+$	anti Sigma plus	SIGMABAR+
PgSz	Σ^0	Sigma zero	SIGMA0
PagSz	$\overline{\Sigma}^0$	anti Sigma zero	SIGMABAR0
PgSa	$\Sigma(1385)P_{13}$	Sigma (1385) P 13	not available
PgSb	$\Sigma(1660)P_{11}$	Sigma (1660) P 11	not available
PgSc	$\Sigma(1670)D_{13}$	Sigma (1670) D 13	not available
PgSd	$\Sigma(1750)S_{11}$	Sigma (1750) S 11	not available
PgSe	$\Sigma(1775)D_{15}$	Sigma (1775) D 15	not available
PgSf	$\Sigma(1915)F_{15}$	Sigma (1915) F 15	not available
PgSg	$\Sigma(1940)D_{13}$	Sigma (1940) D 13	not available

Table B.1: PEN names (*continued*)

PEN	symbol	conventional name	computer name
PgSh	$\Sigma(2030)\mathrm{F}_{17}$	Sigma (2030) F 17	not available
PgSi	$\Sigma(2050)$	Sigma (2250)	not available
Ξ Baryons (S=-2, I=1/2)			
PgXz	Ξ^0	Xi zero	XI0
PagXz	$\overline{\Xi}^0$	anti Xi zero	XIBAR0
PgXm	Ξ^-	Xi minus	XI-
PagXp	$\overline{\Xi}^+$	anto Xi plus	XIBAR+
PgXa	$\Xi(1530)\mathrm{P}_{13}$	Xi (1530) P 13	not available
PgXb	$\Xi(1690)$	Xi (1690)	not available
PgXc	$\Xi(1820)\mathrm{D}_{13}$	Xi (1820) D 13	not available
PgXd	$\Xi(1950)$	Xi (1950)	not available
PgXe	$\Xi(2030)$	Xi (2030)	not available
Ω Baryons (S=-3, I=0)			
PgOm	Ω^-	Omega minus	OMEGA-
PagOp	$\overline{\Omega}^+$	anto Omega plus	OMEGABAR+
PgOma	$\Omega(2250)^-$	Omega (2250) minus	OMEGA(2250)-
Charmed Baryons (C=$+1$)			
PcgLp	Λ_{c}^+	charmed Lambda plus	LAMBDA/C+
PcgXz	Ξ_{c}^0	charmed Xi zero	not available
PcgXp	Ξ_{c}^+	charmed Xi plus	not available
PcgS	$\Sigma_{\mathrm{c}}(2455)$	charmed Sigma 2455	not available
Supersymmetric Particles			
PSgg	$\widetilde{\gamma}$	photino	PHOTINO
PSgxz	$\widetilde{\chi}_{\mathrm{i}}^0$	neutralino	NEUTRALINO
PSZz	$\widetilde{\mathrm{Z}}^0$	supersymmetric Z zero	ZINO
PSHz	$\widetilde{\mathrm{H}}_{\mathrm{j}}^0$	Higgsino	HIGGSINO
PSgxpm	$\widetilde{\chi}^{\pm_{\mathrm{i}}}$	chargino	CHARGINO
PSWpm	$\widetilde{\mathrm{W}}^\pm$	supersymmetric W +-	not available
PSHpm	$\widetilde{\mathrm{H}}^{\pm_{\mathrm{j}}}$	charged Higgsino	not available
PSgn	$\widetilde{\nu}$	scalar neutrino	not available
PSe	$\widetilde{\mathrm{e}}$	scalar electron	not available
PSgm	$\widetilde{\mu}$	scalar muon	not available
PSgt	$\widetilde{\tau}$	scalar tau	not available
PSq	$\widetilde{\mathrm{q}}$	scalar quark	SQUARK
PaSq	$\widetilde{\overline{\mathrm{q}}}$	scalar anti quark	SQUARKBAR
PSg	$\widetilde{\mathrm{g}}$	gluino	GLUINO

The latest PostScript version of this table is in the directory `/usr/local/share/texmf/doc/latex/pennames`.

Appendix C: The TEX Directory Structure (tds)

C.1 Introduction

TEX is a powerful, flexible typesetting system used by tens of thousands of people around the world. It is extremely portable and runs on virtually all operating systems. One unfortunate side effect of TEX's flexibility, however, is that there has been no single "right" way to install it. This has resulted in many sites having different installed arrangements.

A working group of specialists of the TEX world was created in 1994, and it produced recently a proposal (for a complete description see TUGboat 16 (16), pages 401-413, or in electronic form the URL `http://wwwcn.cern.ch/asdoc/WWW/tds-_html/tds_1.html`, whose main points and effects for the CERN setup are the subject of this article.

As a first principle, the TEX Directory Structure (TDS) setup has been designed to work on all modern systems, in particular it copes with MacOS, MS-DOS, OS/2, Unix, VMS, and Windows (3.x, 95, and NT). The same file base can thus be used on multiple platforms simultaneously.

C.2 Top-level directories

The directories under the `TEXMF` root (at CERN we have chosen a GNU-compatible `/usr/local/share/texmf` root directory) identify the major components of a TEX system (see Table C.1).

The top-level directories specified by the TDS are:

`tex` TEX files;

`fonts` font-related files;

`metafont` (non-font) METAFONT files;

`metapost` METAPOST files;

`bibtex` BIBTEX files;

`doc` user documentation;

`source` sources, e.g., `Web2c` sources go in `texmf/source/web2c` or LATEX `dtx` sources go in `texmf/source/latex`;

`implementation` implementations (such as `emtex`, `web2c`); this directory contains files that cannot be shared between implementations, such as pool files (`tex.pool`) and memory dump files (`plain.fmt`), as well as implementation-wide configuration files;

`program` program-specific input files for TEX-related programs (e.g. for `mft`, `dvips`).

C.2.1 A few words on fonts

Font files are stored in separate directories, segregated by file type, font supplier, and typeface.

`texmf/fonts/`*`type/supplier/typeface/`*

type is the type of font file. The TDS reserves the following *type* names, such as `pk` (packed bitmap files), `source` (font sources, i.e. METAFONT files, property lists, etc.), `tfm` (TEX font metric files), `type1` (Type 1 fonts in any format), `vf` (virtual fonts).

supplier is a name identifying font source (examples: `adobe`, `ams`, `public`).

typeface name of a typeface family (examples: `cm`, `euler`, `latex`, `times`).

Some concrete examples:

```
texmf/fonts/source/public/pandora/pnr10.mf
texmf/fonts/tfm/public/cm/cmr10.tfm
texmf/fonts/type1/adobe/utopia/putr.pfa
```

Font bitmap files require two characteristics in addition to the above to be uniquely identifiable: (1) the type of device (i.e., mode) for which the font was created; (2) the resolution of the bitmap.

As TDS has to cope with file systems which do not support long filenames, the font bitmaps are segregated into directories by resolution (e.g., `dpi300/cmr10.pk`), so that we have two more subdirectory levels under `pk` and `gf`:

`texmf/fonts/pk/`*mode*`/`*supplier*`/`*typeface*`/dpi`*nnn*`/`
`texmf/fonts/gf/`*mode*`/`*supplier*`/`*typeface*`/dpi`*nnn*`/`

mode is a name which identifies the device type (examples: `cx`, `gsftopk`, `ljfour`).
`dpi`*nnn* specifies the resolution of the font (examples: `dpi300`, `dpi329`). '`dpi`' stands for dots per inch, i.e., pixels per inch.

C.2.2 Documentation

Most packages come with some form of documentation: user manuals, example files, programming guides, etc. In addition, many independent files not part of any macro or other package describe various aspects of the TEX system.

The TDS specifies that these additional documentation files shall be stored in a structure that parallels to some extent the `fonts` and `tex` directories, as follows:

`texmf/doc/`*category*`/...`

category identifies the general topic of documentation that resides below it; for example, a TEX format name (`latex`), program name (`bibtex`, `tex`), or other system component (`web`, `fonts`).

See Table C.2 for a summary.

C.2.3 Executable programs

The location of executable programs is generally site dependent and also differs across the various platforms. For instance, on Unix, executables are often stored in the directory `/usr/local/bin`.

C.3 Subdirectory searching

Recursive subdirectory searching is the ability to specify a search not only of a specified directory *dir*, but recursively of all directories below *dir*. Different implementations or applications specify subdirectory searching differently, as shown below.

`dvips:` `TEXFONTS_SUBDIR` environment variable;

`kpathsea:` `texmf/subdir//`;

`VMS:` `texmf:[subdir...]`.

On the Unix systems at CERN, we use the Web2c implementation for the TEX programs. All implementation-dependent TEX system files (`.pool`, `.fmt`, `.base`, `.mem`) are stored by default directly in `texmf/web2c`. The configuration file `texmf.cnf` and various subsidiary `MakeTeX...` scripts used as subroutines are also stored there.

Non-TEX specific files are stored following the GNU coding standards. Given a root directory *prefix* (`/usr/local` by default), we have default locations as follows:

`<prefix>/`	installation root
`. bin/`	executables
`. man/`	man pages
`. info/`	info files
`. lib/`	libraries (`libkpathsea.*`)
`. share/`	
`. . texmf/`	TDS root
`. . . web2c/`	implementation-dependent files (`.pool`, `.fmt`, `texmf.cnf`, etc.)

`bibtex/`	BIBTEX input files
`. bib/`	BIBTEX databases
`. . base/`	base distribution (e.g., `xampl.bib`)
`. . misc/`	single-file databases
`. <package>/`	name of a package
`. bst/`	BIBTEX style files
`. . base/`	base distribution (e.g., `plain.bst`, `acm.bst`)
`. . misc/`	single-file styles
`. <package>/`	name of a package
`doc/`	see Table 1.2
`fonts/`	font-related files
`. <type>/`	file type (e.g., `pk`)
`. . <mode>/`	type of output device (for `pk` and `gf` only)
`. . . <supplier>/`	name of a font supplier (e.g., `public`)
`. . . . <typeface>/`	name of a typeface (e.g., `cm`)
`. . . . . dpi<nnn>/`	font resolution (for `pk` and `gf` only)
`<implementation>/`	TEX implementations, by name (e.g., `emtex`)
`metafont/`	METAFONT (non-font) input files
`. base/`	base distribution (e.g., `plain.mf`)
`. misc/`	single-file packages (e.g., `modes.mf`)
`. <package>/`	name of a package (e.g., `mfpic`)
`metapost/`	METAPOST input and support files
`. base/`	base distribution (e.g., `plain.mp`)
`. misc/`	single-file packages
`. <package>/`	name of a package
`. support/`	support files for METAPOST-related utilities
`mft/`	`MFT` inputs (e.g., `plain.mft`)
`<program>/`	TEX-related programs, by name (e.g., `dvips`)
`source/`	program source code by name (e.g., `latex`, `web2c`)
`tex/`	TEX input files
`. <format>/`	name of a format (e.g., `plain`)
`. . base/`	base distribution for format (e.g., `plain.tex`)
`. . misc/`	single-file packages (e.g., `webmac.tex`)
`. . local/`	local additions to or local configuration files for *format*
`. . <package>/`	name of a package (e.g., `graphics`, `mfnfss`)
`. generic/`	format-independent packages
`. . hyphen/`	hyphenation patterns (e.g., `hyphen.tex`)
`. . images/`	image input files (e.g., Encapsulated PostScript)
`. . misc/`	single-file format-independent packages (e.g., `null.tex`)
`. . <package>/`	name of a package (e.g., `babel`)

Table C.1: A skeleton of a TDS `texmf` directory tree

`ams/`	
`. amsfonts/`	`amsfonts.faq`, `amfndoc`
`. amslatex/`	`amslatex.faq`, `amsldoc`
`. amstex/`	`amsgauide`, `joyerr`
`bibtex/`	BibTeX
`. base/`	`btxdoc.tex`
`fonts/`	
`. fontname/`	*Filenames for TeX fonts*
`. oldgerm/`	`corkpapr`
`<format>/`	name of a TeX format (e.g., `generic`, `latex`)
`. base/`	for the base distribution
`. misc/`	for contributed single-file package documentation
`. <package>/`	for *package*
`general/`	across programs, generalities
`. errata/`	`errata`, `errata[1-8]`
`. texcomp/`	*Components of TeX*
`generic/`	for non-format-specific TeX packages
`. babel/`	
`. german/`	`germdoc`
`help/`	meta-information
`. ctan/`	info about CTAN mirror sites
`. faq/`	FAQs of `comp.text.tex`, etc.
`html/`	HTML files
`info/`	GNU Info files, made from Texinfo sources
`latex/`	example of *format*
`. base/`	`ltnews*`, `*guide`, etc.
`. graphics/`	`grfguide`
`<program>/`	TeX-related programs, by name (examples follow)
`metafont/`	`mfbook.tex`, `metafont-for-beginners`, etc.
`metapost/`	`mpman`, `manfig`, etc.
`tex/`	`texbook.tex`, *A Gentle Introduction to TeX*, etc.
`web/`	`webman`, `cwebman`

Table C.2: A skeleton of a TDS directory tree under `texmf/doc`

Index