1 December 1994

Contents

1 Introduction

This document describes the new font selection features of the LaTeX Document Preparation System. It is intended for package writers who want to write font-loading packages similar to ntimesn or nlatexsymn. This document is only a brief introduction to the new facilities and is intended for package writers who are familiar with TeX fonts and LaTeX packages. It is *neither* a user-guide *nor* a reference manual for fonts in LaTeX2e.

1.1 LaTeX2e fonts

The most important difference between LaTeX 2.09 and LaTeX2e is the way that fonts are selected. In LaTeX 2.09, the Computer Modern fonts were built into the LaTeX format, and so customizing LaTeX to use other fonts was a major effort.

In LaTeX2e, very few fonts are built into the format, and there are commands to load new text and math fonts. Packages such as ntimesn or nlatexsymn allow authors to access these fonts. This document describes how to write similar font-loading packages.

The LaTeX2e font selection system was first released as the iNew Font Selection Schemej (NFSS) in 1989, and then in release 2 in 1993. LaTeX2e includes NFSS release 2 as standard.

1.2 Overview

This document contains an overview of the new font commands of LaTeX.

Section ? describes the commands for selecting fonts in classes and packages. It lists the five LaTeX font attributes, and lists the commands for selecting fonts. It also describes how to customize the author commands such as nn and n*n* to suit your document design.

Section ? explains the commands for controlling LaTeX math fonts. It describes how to specify new math fonts and new math symbols.

Section ? explains how to install new fonts into LaTeX. It shows how LaTeX font attributes are turned into TeX font names, and how to specify your own fonts using font definition files.

Section ? discusses text font encodings. It describes how to declare a new encoding and how to define commands, such as nÆn or n.n, which have different definitions in different encodings, depending on whether ligatures, etc. are available in the encoding.

Section ? covers font miscellania. It describes how LaTeX performs font substitution, how to customize fonts that are preloaded in the LaTeX format, and the naming conventions used in LaTeX font selection.

1.3 Further information

For a general introduction to LaTeX, including the new features of LaTeX2e, you should read , Leslie Lamport, Addison Wesley, 2nd ed, 1994.

A more detailed description of the LaTeX font selection scheme is to be found in , Goossens, Mittelbach and Samarin, Addison Wesley, 1994.

The LaTeX font selection scheme is based on TeX, which is described by its developer in *The* TeX*book*, Donald E. Knuth, Addison Wesley, 1986, revised in 1991 to include the features of TeX 3.

Sebastian Rahtzjs npsnfssn software contains the software for using a large number of Type 1 fonts (including the Adobe Laser Writer 35 and the Monotype CD-ROM fonts) in LaTeX. It should be available from the same source as your copy of LaTeX.

The npsnfssn software uses fonts generated by Alan Jeffreyjs nfontinstn software. This can convert fonts from Adobe Font Metric format into a format readable by LaTeX, including the generation of the n.fdn files described in Section ?. The nfontinstn software should be available from the same source as your copy of LaTeX.

Whenever practical, LaTeX uses the font naming scheme described in *Filenames for fonts*, Karl Berry, *TUGboat* 11(4), 1990.

The class-writerjs guide describes the new LaTeX features for writers of document classes and packages and is kept in nclsguide.texn.

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

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

2 Text fonts

This section describes the commands available to class and package writers for specifying and selecting fonts.

2.1 Text font attributes

Every text font in LaTeX has five *attributes*:

encoding This specifies the order that characters appear in the font. The two most common text encodings used in LaTeX are Knuthjs iTeX textj encoding, and the iTeX text extendedj encoding developed by the TeX Users Group members during a TeX Conference at Cork in 1990 (hence its informal name iCork encodingj).

family The name for a collection of fonts, usually grouped under a common name by the font foundry. For example, iAdobe Timesj, iITC Garamondj, and Knuthjs iComputer Modern Romanj are all font families.

series How heavy or expanded a font is. For example, imedium weightj, inarrowj and ibold extendedj are all series.

shape The form of the letters within a font family. For example, iitalicj, iobliquej and iuprightj (sometimes called iromanj) are all font shapes.

size The design size of the font, for example i10ptj.

The possible values for these attributes are given short acronyms by LaTeX. The most common values for the font encoding are:

nOT1n TeX text
nT1n TeX extended text

nOMLnTEX math italic
nOMSnTEX math symbols
nOMXnTEX math large symbols
nUnUnknown
nL<xx>nA local encoding

The ilocalj encodings are intended for font encodings which are only locally available, for example a font containing a university logo in various sizes. There are far too many font families to list them all, but some common ones are:

ncmrnComputer Modern Roman
ncmssnComputer Modern Sans
ncmttnComputer Modern Typewriter
ncmmnComputer Modern Math Italic
ncmsynComputer Modern Math Symbols
ncmexnComputer Modern Math Extensions
nptmnAdobe Times
nphvnAdobe Helvetica
npcrnAdobe Courier

The most common values for the font series are:

nmnMedium
nbnBold
nbxnBold extended
nsbnSemi-bold
ncnCondensed

The most common values for the font shape are:

nnnNormal (that is iuprightj or iromanj)
nitnItalic
nslnSlanted (or iobliquej)
nscnCaps and small caps

The font size is specified as a dimension, for example n10ptn or n1.5inn or n3mmn. These five parameters specify every LaTeX font, for example:

LaTeX specification					*Font*	TeX *font name*
ncmrnmmnnnnn10ptn	nOT1n				Computer Modern Roman 10pt	ncmr10n
ncmssnmmnslnn12ptn	nOT1n				Computer Modern Sans Oblique 12pt	ncmssi12n
ncmmnmmnitnn10ptn					Computer Modern Math	ncmmi10n

nOMLn Italic 10pt
nptmmnbnnitmn18ptnAdobe Times Bold Italic nptmbq at 18ptn
nT1n 18pt

These five parameters are displayed whenever LaTeX gives an overfull box warning, for example:

```
   Overfull \hbox (3.80855pt too wide) in paragraph at
lines 314--318
   []\OT1/cmr/m/n/10 Normally [] and [] will be iden-ti-
cal,
```

The author commands for fonts set the five attributes:

Author command	*Attribute*	*Value in* narticlen *class*
n..n or nn	family	ncmrn
n..n or nn	family	ncmssn
n..n or nn	family	ncmttn
n..n or nn	series	nmn
n**..**n or n**n**	series	nbxn
n..n or nn	shape	nnn
n..n or n*n*	shape	nitn
n..n or n*n*	shape	nsln
n..n or nn	shape	nscn
nn	size	n5ptn
nn	size	n7ptn
nn	size	n8ptn
nn	size	n9ptn
nn	size	n10ptn
nn	size	n12ptn
nn	size	n14.4ptn
nn	size	n17.28ptn
nn	size	n20.74ptn
nn	size	n24.88ptn

The values used by these commands are determined by the document class, using the parameters defined in Section ?.

Note that there are no author commands for selecting new encodings. These should be provided by packages, such as the nfontencn package.

This section does not explain how LaTeX font specifications are turned into TeX font names. This is described in Section ?.

2.2 Selection commands

The low-level commands used to select a text font are as follows.

[Sorry. Ignored `\begindecl ... \enddecl`]

Each of these commands sets one of the font attributes; nn also sets nn. The actual

font in use is not altered by these commands, but the current attributes are used to determine which font to use after the next nn command.

[Sorry. Ignored `\begindecl ... \enddecl`]

Selects a text font, based on the current values of the font attributes.

Warning: there *must* be a nn command immediately after any settings of the font parameters by (some of) the five n<parameter>n commands, before any following text. For example, it is legal to say:

```
\fontfamilyptm\fontseriesb\selectfont Some text.
```

but it is *not* legal to say:

```
\fontfamilyptm Some \fontseriesb\selectfont text.
```

You may get unexpected results if you put text between a n<parameter>n command and a nn.

[Sorry. Ignored `\begindecl ... \enddecl`]

A short hand for the equivalent nn? commands followed by a call to nn.

2.3 Internals

The current values of the font attributes are held in internal macros.

[Sorry. Ignored `\begindecl ... \enddecl`]

These hold the current values of the encoding, the family, the series, the shape, the size, the baseline skip, the main math size, the script math size and the scriptscript math size. The last three are accessible only within a formula; outside of math they may contain arbitrary values.

For example, to set the size to 12 without changing the baseline skip:

```
\fontsize12\f@baselineskip
```

However, you should *never* alter the values of the internal commands directly; they must only be modified using the low-level commands like nn, nn, etc. If you disobey this warning you might produce code that loops.

2.4 Parameters for author commands

The parameter values set by author commands such as nn and nn, etc. are not hard-wired into LaTeX; instead these commands use the values of a number of parameters set by the document class and packages. For example, nn is the name of the default family selected by nn and nn. Thus to set a document in Adobe Times, Helvetica and Courier, the document designer specifies:

```
\renewcommand\rmdefaultptm
\renewcommand\sfdefaultphv
\renewcommand\ttdefaultpcr
```

[Sorry. Ignored `\begindecl ... \enddecl`]

The encoding, family, series and shape of the main body font. By default these are nOT1n, nn, nmn and nnn. Note that since the default family is nn, this means that changing nn will change the main body font of the document.

[Sorry. Ignored `\begindecl ... \enddecl`]

The families selected by nn, nn, nn, nn, nn and nn. By default these are ncmrn, ncmssn and ncmttn.

[Sorry. Ignored `\begindecl ... \enddecl`]

The series selected by n**n**, n**n, n**n **and n**n. By default these are nbxn and nmn. These values are suitable for the default families used. If other fonts are used as standard document fonts, for example, some PostScript fonts, it might be necessary to adjust the value of nn to nbn since only a few such families have ibold extendedj series. An alternative (taken for the fonts provided by npsnfssn) is to define silent substitutions from nbxn series to nbn series with special nn declarations and the nssubn size function, see Section ?.

[Sorry. Ignored \begindecl ... \enddecl]

The shapes selected by n*n*, n*nn, nn, nn, nn, nn, n* n *AND n* n. By default these are nitn, nsln, nscn and nnn.

Note that there are no parameters for the size commands. These should be defined directly in class files, for example:

```
\renewcommand\normalsize\fontsize1012\selectfont
```

More elaborate examples (setting additional parameters when the text size is changed) can be found in nclasses. dtxn the source documentation for the classes narticlen, nreportn, and nbookn.

2.5 Special font declaration commands

[Sorry. Ignored \begindecl ... \enddecl]

Declares command cmd to be a font switch which selects the font that is specified by the attributes ENC, family, series, shape, and size.

The font is selected without any adjustments to baselineskip and other surrounding conditions.

This example makes n.n select a small dot very quickly:

```
\DeclareFixedFont\picturecharOT1cmrmn5
```

[Sorry. Ignored \begindecl ... \enddecl]

Declares command cmd to be a font command with one argument. The current font attributes are locally modified by font-switches and then the argument of cmd is typeset in the resulting new font.

Commands defined by nn automatically take care of any necessary italic correction (on either side).

The following example shows how nn is defined by the kernel.

```
\DeclareTextFontCommand\textrm\rmfamily
```

To define a command that always typeset its argument in the italic shape of the main document font you could declare:

```
\DeclareTextFontCommand\normalit\normalfont\itshape
```

This declaration can be used to change the meaning of a command; if cmd is already defined, a log that it has been redefined is put in the transcript file.

[Sorry. Ignored \begindecl ... \enddecl]

Declares command cmd to be a font switch (i.e. used with the syntax n<cmd>...n) having the definition text-switch when used in text and the definition math-switch when used in a formula. Math alphabet commands, like n*n*, when used within math-switch should not have an argument. Their use in this argument causes their semantics to change so that they here act as a font switch, as required by the usage of the cmd.

This declaration is useful for setting up commands like nn to behave as they did in LaTeX 2.09. We strongly urge you *not* to misuse this declaration to invent new font commands.

The following example defines n*n to produce the italic shape of the main document font if used in text and to switch to the font that would normally be produced by the math alphabet nn if used in a formula.*

```
\DeclareOldFontCommand\it\normalfont\itshape\mathit
```

This declaration can be used to change the meaning of a command; if cmd is already defined, a log that it has been redefined is put in the transcript file.

3 Math fonts

This section describes the commands available to class and package writers for specifying math fonts and math commands.

3.1 Math font attributes

The selection of fonts within math mode is quite different to that of text fonts.

Some math fonts are selected explicitly by one-argument commands such as n**max**n or n**vec**n: these are called *math alphabets*. These math alphabet commands affect only the font used for letters and symbols of type nn (see Section ?); other symbols within the argument will be left unchanged. The predefined math alphabets are:

Alphabet	*Description*	*Example*
nn	default	*abcXYZ*
nn	roman	*abcXYZ*
n**n**	bold roman	**abcXYZ**
nn	sans serif	*abcXYZ*
n*n*	text italic	*abcXYZ*
nn	typewriter	*abcXYZ*
nn	calligraphic	XYZ

Other math fonts are selected implicitly by TeX for symbols, with commands such as n*n (producing *) or with straight characters like n»n or n+n. Fonts containing such math symbols are called *math symbol fonts*. The predefined math symbol fonts are:

Symbol font	*Description*	*Example*

noperatorsn	symbols from nn	[+]
nlettersn	symbols from nn	<<*>>
nsymbolsn	most LaTeX symbols	*
nlargesymbolsn	large symbols	

Some math fonts are both *math alphabets* and *math symbol fonts*, for example nn and noperatorsn are the same font, and nn and nlettersn are the same font.

Math fonts in LaTeX have the same five attributes as text fonts: encoding, family, series, shape and size. However, there are no commands that allow the attributes to be individually changed. Instead, the conversion from math fonts to these five attributes is controlled by the *math version*. For example, the nnormaln math version maps:

Math font		*External font*			
Alphabets	*Symbol fonts*	*Attributes*			
nnnlettersn		nOMLnncmmn	nmn	nitn	
nnnoperatorsn		nOT1n ncmrn	nmn	nnn	
nnnsymbolsn		nOMSnncmsyn	nmn	nnn	
	nlargesymbolsn	nOMXncmexn	nmn	nnn	
n**n**		nOT1n ncmrn	nbxn	nnn	
nn		nOT1n ncmssn	nmn	nnn	
n*n*		nOT1n ncmrn	nmn	nitn	
nn		nOT1n ncmttn	nmn	nnn	

The nboldn math version is similar except that it contains bold fonts. The command nn selects the nboldn math version.

Math versions can only be changed outside of math mode.

The two predefined math versions are:

nnormaln	the default math version
nboldn	the bold math version

Packages may define new math alphabets, math symbol fonts, and math versions. This section describes the commands for writing such packages.

3.2 Selection commands

There are no commands for selecting symbol fonts. Instead, these are selected indirectly through symbol commands like n*n. Section ? explains how to define symbol commands.

[Sorry. Ignored `\begindecl ... \enddecl`]

Each math alphabet is a command which can only be used inside math mode. For example, n*x*+*y*+Zn produces *x*+*y*+Z.

[Sorry. Ignored `\begindecl ... \enddecl`]

This command selects a math version; it can only be used outside math mode. For example, nn is defined to be nboldn.

3.3 Declaring math versions

[Sorry. Ignored `\begindecl ... \enddecl`]

Defines version-name to be a math version.

The newly declared version is initialised with the defaults for all symbol fonts and math alphabets declared so far (see the commands nn and nn).

If used on an already existing version, an information message is written to the transcript file and all previous nn or nn declarations for this version are overwritten by the math alphabet and symbol font defaults, i.e. one ends up with a virgin math version.

Example:

```
\DeclareMathVersionnormal
```

3.4 Declaring math alphabets

[Sorry. Ignored `\begindecl ... \enddecl`]

Defines math-alph to be a new math alphabet.

The arguments encoding family series shape are the default values for this math alphabet in all math versions; these can be reset later for a particular math version by a nn command. If shape is empty then the math-alph is declared to be invalid in all versions, unless it is set by a later nn command.

Checks that math-alph can be used and that encoding is a valid encoding scheme.

In these examples, nn is defined everywhere but nn, by default, is defined nowhere.

```
\DeclareMathAlphabet\fooOT1cmttmn
\DeclareMathAlphabet\bazOT1
```

[Sorry. Ignored `\begindecl ... \enddecl`]

Changes, or sets, the font for the math alphabet math-alph in math version version-name to encodingfamilyseriesshape.

Checks that math-alph is a math alphabet, version-name is a math version and encoding is a known encoding scheme.

This example defines nn for the inormalj math version only:

```
\SetMathAlphabet\baznormalOT1cmssmn
```

Note that this declaration is not used for all math alphabets: Section ? describes nn, which is used to set up math alphabets contained in fonts which have are declared as symbol fonts.

3.5 Declaring symbol fonts

[Sorry. Ignored `\begindecl ... \enddecl`]

Defines sym-font-name to be a new symbol font.

The arguments encoding family series shape are the default values for this symbol font in all math versions; these can be reset later for a particular math version by a nn command.

Checks that sym-font-name can be used and that encoding is a declared encoding scheme.

For example, the following sets up the first four standard math symbol fonts:

```
\DeclareSymbolFontoperatorsOT1cmrmn
\DeclareSymbolFontlettersOMLcmmmit
\DeclareSymbolFontsymbolsOMScmsymn
\DeclareSymbolFontlargesymbolsOMXcmexmn
```

[Sorry. Ignored `\begindecl ... \enddecl`]

Changes the symbol font sym-font-name for math version version name to encoding family series shape.

Checks that sym-font-name is a symbol font, version name is a known math version and encoding is a declared encoding scheme.

For example, the following come from the set up of the iboldj math version:

```
\SetSymbolFontoperatorsboldOT1cmrbxn
\SetSymbolFontlettersboldOMLcmmbit
```

[Sorry. Ignored `\begindecl ... \enddecl`]

Allows the previously declared symbol font sym-font-name to be also the math alphabet id (in all math versions).

This declaration should be used in preference to nn and nn when a math alphabet is the same as a symbol font; this is because it makes better use of the limited number (only 16) of TEXjs math groups.

Checks that math-alph can be defined and that sym-font-name is a symbol font.

Example:

```
\DeclareSymbolFontAlphabet\mathrmoperators
\DeclareSymbolFontAlphabet\mathcalsymbols
```

3.6 Declaring math symbols

[Sorry. Ignored `\begindecl ... \enddecl`]

The symbol can be either a single character such as in»nj, or a macro name, such as n \i \su(,,)n.

Defines the symbol to be a math symbol of type type in slot slot of symbol font sym-font-name. The type can be given as a number or as a command:

Type	*Meaning*	*Example*
n0n or nn	Ordinary	*
n1n or nn	Large operator	
n2n or nn	Binary operation	*
n3n or nn	Relation	*
n4n or nn	Opening	*
n5n or nn	Closing	*
n6n or nn	Punctuation	;
n7n or nn	Alphabet character	*A*

Only symbols of type nn will be affected by math alphabet commands: within the argument of a math alphabet command they will produce the character in slot slot of that math alphabetjs font. Symbols of other types will always produce the same symbol (within one math version).

nn allows a macro symbol to be redefined only if it was previously defined to be a math symbol. It also checks that the sym-font-name is a declared symbol font.

Example:

```
\DeclareMathSymbol\alpha0letters"0B
\DeclareMathSymbol\lessdot\mathbinAMSb"0C
\DeclareMathSymbol\foo\mathalphaAMSb"0C
```

[Sorry. Ignored `\begindecl ... \enddecl`]

Defines cmd to be a math delimiter where the small variant is in slot slot-1 of symbol font sym-font-name-1 and the large variant is in slot slot-2 of symbol font sym-font-name-2. Both symbol fonts must have been declared previously.

Checks that sym-font-name-i are both declared symbol fonts.

If TEX is not looking for a delimiter, cmd is treated just as if it had been defined with nn using type,sym-font-name-1 and slot-1. In other words, if a command is defined as a delimiter then this automatically defines it as a math symbol.

Example:

```
\DeclareMathDelimiter\langle\mathopensymbols"68
 largesymbols"0A
```

[Sorry. Ignored `\begindecl ... \enddecl`]

Defines cmd to act as a math accent.

The accent character comes from slot slot in sym-font-name. The type can be either nn or nn; in the latter case the accent character changes font when used in a math alphabet.

Example:

```
\DeclareMathAccent\acute\mathalphaoperators"13
\DeclareMathAccent\vec\mathordletters"7E
```

[Sorry. Ignored `\begindecl ... \enddecl`]

Defines cmd to be a radical where the small variant is in slot slot-1 of symbol font sym-font-name-1 and the large variant is in slot slot-2 of symbol font sym-font-name-2. Both symbol fonts must have been declared previously.

Example (probably the only use for it!):

```
\DeclareMathRadical\sqrtsymbols"70largesymbols"70
```

3.7 Declaring math sizes

[Sorry. Ignored `\begindecl ... \enddecl`]

Declares that mt-size is the math text size, s-size is the script size and ss-size the scriptscript size to be used in math, when t-size is the current text size. For text sizes for which no such declaration is given the script and scriptscript size will be calculated and then fonts are loaded for the calculated sizes or the best approximation (this may result in a warning message).

Normally, t-size and mt-size will be identical; however, if, for example, PostScript text fonts are mixed with bit-map math fonts then you may not have available a mt-size for every t-size.

Example:

```
\DeclareMathSizes13.8212.4107
```

4 Font installation

This section explains how LATEXjs font attributes are turned into TEX font specifications.

4.1 Font definition files

The description of how LaTeX font attributes are turned into TeX fonts is usually kept in a *font definition* (n.fdn) file. The n.fdn file for family family in encoding encoding is called n<ENC><family>.fdn, for example nOT1cmr.fdn for Computer Modern Roman or nT1ptm.fdn for Adobe Times.

Whenever LaTeX encounters an encoding/family combination that it does not know (e.g. if the document designer says nptmn) then LaTeX attempts to load the appropriate n.fdn file. kNot knownl means: there was no nn declaration issued for this encoding/family combination.

The declarations in the n.fdn file are responsible for telling LaTeX how to load fonts for that encoding/family combination. If the n.fdn file could not be found, a warning is issued and font substitutions are made.

4.2 Font definition file commands

The n.fdn files should contain only commands from this section. Note that those commands can also be used outside a n.fdn file: they can be put in package or class files, or even in the preamble of a document.

[Sorry. Ignored `\begindecl ... \enddecl`]

The n.fdn file should announce itself with a nn command, as described in . For example:

```
\ProvidesFileT1ptm.fd[1994/06/01 Adobe Times font
definitions]
```

[Sorry. Ignored `\begindecl ... \enddecl`]

Declares a font family family to be available in encoding scheme encoding.

The loading-settings are executed immediately after loading any font with this encoding and family.

Checks that encoding was previously declared.

This example refers to the Computer Modern Typewriter font family in the Cork encoding:

```
\DeclareFontFamilyT1cmtt\hyphenchar\font=-1
```

Each n.fdn file should contain exactly one nn command, and it should be for the appropriate encoding/family combination.

[Sorry. Ignored `\begindecl ... \enddecl`]

Declares a font shape combination; here loading-info contains the information that combines sizes with external fonts. The syntax is complex and is described in Section ? below.

The loading-settings are executed after loading any font with this font shape. They are executed immediately after the iloading-settingsj which were declared by nn and so they can be used to overwrite the settings made at the family level.

Checks that the combination encodingfamily was previously declared via nn.

Example:

```
\DeclareFontShapeOT1cmrmsl%
<<5-8>> sub * cmr/m/n
<<8>> cmsl8
<<9>> cmsl9
<<10>> <<10.95>> cmsl10
<<12>> <<14.4>> <<17.28>> <<20.74>> <<24.88>> cmsl12
```

An n.fdn file can contain any number of nn commands, which should be for the appropriate encoding and family.

4.3 Font file loading information

The information which tells LaTeX exactly which font (`.tfm`) files to load is contained in the loading-info part of a nn declaration. This part consists of one or more fontshape-decls, each of which has the following form:

fontshape-decl size-infos font-info
size-infos size-info | size-infos size-info
size-info kn«nl number-or-range kn»nl
font-info [size-function kn*nl] [kn[nl optarg kn]nl] fontarg

The number-or-range denotes the size or size-range for which this entry applies. If it contains a hyphen it is a range: lower bound on the left (if missing, zero implied), upper bound on the right (if missing, * implied). For ranges, the upper bound is *not* included in the range and the lower bound is.

Examples:

n«10»n	simple size	10pt only
n«-8»n	range	all sizes less than 8pt
n«8-14.4»n	range	all sizes greater than or equal to 8pt but less than 14.4pt
n«14.4-»n	range	all sizes greater than or equal 14.4pt

If more than one size-info entry follows without any intervening font-info, they all share the next font-info.

The size-function, if present, handles the use of font-info. If not present, the iemptyj size-function is assumed.

All the size-infos are inspected in the order in which they appear in the font shape declaration. If a size-info matches the requested size, its size-function is executed. If n@fontn is non-empty afterwards this process stops, otherwise the next size-info is inspected. (See also nn.)

If this process does not lead to a non-empty n@fontn, LaTeX tries the nearest simple size. If the entry contains only ranges an error is returned.

4.4 Size functions

LaTeX provides the following size functions, whose iinputsj are fontarg and optarg (when present).

ij (empty) Load the external font fontarg at the user-requested size. If optarg is present, it is used as the scale-factor.

s Like the empty function but without terminal warnings, only loggings.

gen Generates the external font from fontarg followed by the user-requested size, e.g. n«8» «9» «10» gen * cmttn

sgen Like the igenj function but without terminal warnings, only loggings.

sub Tries to load a font from a different font shape declaration given by fontarg in the form familyn/nseriesn/nshape.

ssub Silent variant of isubj, only loggings.

subf Like the empty function but issues a warning that it has to substitute the external font fontarg because the desired font shape was not available in the requested size.

ssubf Silent variant of isubfj, only loggings.

fixed Load font fontarg as is, disregarding the user-requested size. If present, optarg gives the kat ?ptl size to be used.

sfixed Silent variant of ifixedj, only loggings.

Examples for the use of most of the above size functions can be found in the file ncmfonts.fddnnthe source for the standard n.fdn files describing the Computer Modern fonts by Donald Knuth.

[Sorry. Ignored `\begindecl ... \enddecl`]

Declares a size-function name for use in nn commands. The interface is still under development but there should be no real need to a define new size functions.

The code is executed when the size or size-range in nn matches the user-requested size.

The arguments of the size-function are automatically parsed and placed into n@argn and n@argn for use in code. Also available, of course, is n@sizen, which is the user-requested size.

To signal success code must define the command n@fontn to contain the external name and any scaling options (if present) for the font to be loaded.

This example sets up the iemptyj size function (simplified):

```
 \DeclareSizeFunction
\edef\external@font\mandatory@arg\space at\f@size
```

5 Encodings

This section explains how to declare and use new font encodings and how to declare commands for use with particular encodings.

5.1 The fontenc package

Users can select new font encodings using the nfontencn package. The nfontencn package has options for encodings; the last option becomes the default encoding. For example, to use the nOT2n (Washington University Cyrillic encoding) and nT1n encodings, with nT1n as the default, an author types:

```
\usepackage[OT2,T1]fontenc
```

This package loads the *encoding definition files* (n<ENC>enc.defn files) for each font encoding ENC given as an option but not already declared; it also sets nn to be the last encoding in the option list.

LaTeX currently predefines the nOT1n and nT1n text encodings, and provides the files nOT1enc.defn and nT1enc.defn. Other encoding set-ups might be added to the distribution at a later stage.

Thus the example above loads the file nOT2enc.defn and sets nn to nT1n.

5.2 Encoding definition files

Warning: Some aspects of the contents of font definition files are still under development. Therefore, the current versions of the files nOT1enc.defn and nT1enc. defn are temporary versions and should not be used as models for producing further

such files. For further information you should read the documentation in nltoutenc. dtxn.

The declarations in the encoding definition file are responsible for declaring the encoding and telling LaTeX how to produce characters in this encoding.

The n<ENC>enc.defn files should contain only commands from this section. As with the font definition file commands, it is also possible (although normally not necessary) to use these declarations directly within a class or package file.

[Sorry. Ignored `\begindecl ... \enddecl`]

The n<ENC>enc.defn file should announce itself with a nn command, described in . For example:

```
\ProvidesFileOT2enc.def
[1994/06/01 Washington Univ. Cyrillic encoding]
```

[Sorry. Ignored `\begindecl ... \enddecl`]

Declares a new encoding scheme encoding.

The text-settings are declarations which are executed every time nn changes the encoding to be encoding.

The math-settings are similar but are for math alphabets. They are executed whenever a math alphabet with this encoding is called.

Spaces within the arguments are ignored to avoid surplus spaces in the document. If a real space is necessary use nn.

Example:

```
\DeclareFontEncodingOT1
```

Some author commands need to change their definition depending on which encoding is currently in use. For example, in the nOT1n encoding, the letter iÆj is in slot n"1Dn, whereas in the nT1n encoding it is in slot n"C6n. So the definition of nÆn has to change depending on whether the current encoding is nOT1n or nT1n. The following commands allow this to happen.

[Sorry. Ignored `\begindecl ... \enddecl`]

This command is like n except that it defines a command which is specific to one encoding. For example, the definition of nån in the nOT1n encoding is:

```
\DeclareTextCommand\aaOT1\accent23a
```

takes the same optional arguments as n

The resulting command is robust, even if the code in definition is fragile.

It does not produce an error if the command has already been defined but logs the redefinition in the transcript file.

[Sorry. Ignored `\begindecl ... \enddecl`]

This command is the same as nn, except that if cmd is already defined in encoding encoding, then the definition is ignored.

[Sorry. Ignored `\begindecl ... \enddecl`]

This command defines a text symbol with slot slot in the encoding. For example, the definition of nßn in the nOT1n encoding is:

```
\DeclareTextSymbol\ssOT125
```

It does not produce an error if the command has already been defined but logs the redefinition in the transcript file.

[Sorry. Ignored `\begindecl ... \enddecl`]

This command declares a text accent, with the accent taken from slot slot in the encoding. For example, the definition of n.n in the nOT1n encoding is:

```
\DeclareTextAccent\"OT1127
```

It does not produce an error if the command has already been defined but logs the redefinition in the transcript file.

[Sorry. Ignored `\begindecl ... \enddecl`]

This command declares that the composite letter formed from applying cmd to letter is defined to be simply slot slot in the encoding. The letter should be a single letter (such as nan) or a single command (such as nin).

For example, the definition of nán in the nT1n encoding could be declared like this:

```
\DeclareTextComposite\'T1a225
```

The cmd should have been previously declared with nn, or as a one-argument nn.

[Sorry. Ignored `\begindecl ... \enddecl`]

This is a more general form of nn, which allows for an arbitrary definition, not just a slot. The main use for this is to allow accents on nin to act like accents on nin, for example:

```
\DeclareTextCompositeCommand\'OT1i\'\i
```

It has the same restrictions as nn.

5.3 Default definitions

1994/12/01 The commands in ENCnenc.defn files allow encoding-specific commands to be defined, but they do not allow commands to be used in other encodings. For example, the OMS encoding contains the symbol i§j, but we need to be able to use the command n§n in any encoding, not just OMS.

To allow this, LaTeX has commands for giving default definitions for commands, which are used when the command is not defined in the current encoding. For example, the default encoding for n§n is OMS, and so in an encoding (such as OT1) which does not contain n§n, the OMS version is selected. But in an encoding (such as T1) which does contain n§n, the version for that encoding is used.

Note these commands should *not* occur in ENCnenc.defn files, since these should only define commands for that encoding. They should instead be placed in packages.

[Sorry. Ignored `\begindecl ... \enddecl`]

This command allows an encoding-specific command to be given a default definition. For example, the default definition for n©n is defined be be a circled icj with:

```
\DeclareTextCommandDefault\copyright\textcircledc
```

[Sorry. Ignored `\begindecl ... \enddecl`]

These commands allow an encoding-specific command to be given a default encoding. For example, the default encoding for n.n and næn is set to be OT1 by:

```
\DeclareTextAccentDefault\"OT1
\DeclareTextSymbolDefault\aeOT1
```

Note that nn can be used on any one-argument encoding-specific command, not just those defined with nn. Similarly, nn can be used on any encoding-specific command with no arguments, not just those defined with nn.

For more examples of these definitions, see nltoutenc.dtxn.

[Sorry. Ignored `\begindecl ... \enddecl`]

This command is the same as nn, except that if the command already has a default definition, then the definition is ignored. This is useful to give ifakedj definitions of symbols which may be given irealj definitions by other packages. For example, a package might give a fake definition of nn by saying:

```
\ProvideTextCommandDefault\textonequarter$\m@th\
frac14$
```

5.4 Encoding defaults

[Sorry. Ignored `\begindecl ... \enddecl`]

Declares text-settings and math-settings for all encoding schemes. These are executed before the encoding scheme dependent ones are executed so that one can use the defaults for the major cases and overwrite them if necessary using nn.

If nn is used as an argument, the current setting of this default is left unchanged.

This example is used by amsfonts.sty for accent positioning; it changes only the math settings:

```
\DeclareFontEncodingDefaults\relax\def\accentclass@7
```

[Sorry. Ignored `\begindecl ... \enddecl`]

Declares the default values for font substitution which will be used when a font with encoding encoding should be loaded but no font can be found with the current attributes.

These substitutions are local to the encoding scheme because the encoding scheme is never substituted! They are tried in the order shape then series and finally family.

If no defaults are set up for an encoding, the values given by nn are used.

The font specification for encodingfamilyseriesshape must have been defined by nn before the n

6 Miscellania

This section covers the remaining font commands of LaTeX.

6.1 Font substitution

[Sorry. Ignored `\begindecl ... \enddecl`]

Declares encodingfamilyseriesshape to be the font shape used in cases where the standard substitution mechanism fails (i.e. would loop). For the standard mechanism see the command nn above.

The font specification for encodingfamilyseriesshape must have been defined by nn before the n

6.2 Preloading

[Sorry. Ignored `\begindecl ... \enddecl`]

Specifies the fonts that should be preloaded by the format. These commands should be put in a npreload.cfgn file, which is read in when the LaTeX format is being built. Read npreload.dtxn for more information on how to built such a configuration file.

Example:

```
\DeclarePreloadSizesOT1cmrmsl10,10.95,12
```

6.3 Naming conventions

Math alphabet commands all start with n...n: examples are n**n**, nn, etc.

The text font changing commands with arguments all start with n...n: e.g. n**n** and nn. The exception to this is n*n*, since it occurs very commonly in author documents and so deserves a shorter name.

Names for encoding schemes are strings of up to three letters, all upper case. The LaTeX3 project reserves the use of encodings starting with nTn (standard 256-long text encodings), nMn (standard 256-long math encodings), nSn (standard 256-long symbol encodings), nOTn (standard 128-long text encodings) and nOMn (standard 128-long math encodings). Please do not use the above starting letters for non-portable encodings. If new standard encoding emerge then we shall add them in a later release of LaTeX.

Encoding schemes which are local to a site should start with nLn.

Font family names should contain up to five lower case letters. Where possible, these should conform to the *Filenames for fonts* font naming scheme.

Font series names should contain up to four lower case letters.

Font shapes should contain up to two letters lower case.

Names for symbol fonts are built from lower and upper case letters with no restriction.

Whenever possible, you should use the series and shape names suggested in since this will make it easier to combine new fonts with existing fonts.

1994/12/01 Where possible, text symbols should be named as nn followed by the Adobe glyph name: for example nn or nn. Similarly, math symbols should be named as nn followed by the glyph name, for example nn or nn. Commands which can be used in text or math can then be defined using nn, for example:

```
   \DeclareRobustCommand\pounds%
 \ifmmode \mathsterling \else \textsterling \fi
```

Note that commands defined in this way must be robust, in case they get put into a section title or other moving argument.

References

[] Michel Goossens, Frank Mittelbach and Alexander Samarin. *The* LaTeX *Companion*. Addison-Wesley, Reading, Massachusetts, 1994.

[] Donald E. Knuth. Typesetting concrete mathematics. *TUGboat*, 10(1): 31m36, April 1989.

[] Leslie Lamport. LaTeX*: A Document Preparation System*. Addison-Wesley, Reading, Massachusetts, second edition, 1994.