

# The kerntest package

Harald Harders  
h.harders@tu-bs.de

Version v1.32 (2004/04/14), printed 14th April 2004

## Abstract

This class makes it easy to generate tables that show many different kerning pairs of an arbitrary font, usable by L<sup>A</sup>T<sub>E</sub>X. It shows the kerning values that are used by the the font by default.

In addition, this class enables the user to alternate the kernings and to observe the results. Kerning pairs can be defined for groups of similar glyphs at once. Automatically, an `mtx` file is generated that can be loaded by `fontinst` to introduce the user-made kernings into the virtual font for L<sup>A</sup>T<sub>E</sub>X.

## Contents

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introduction</b>                            | <b>2</b>  |
| <b>2</b> | <b>Usage of the class</b>                      | <b>3</b>  |
| 2.1      | Introduction . . . . .                         | 3         |
| 2.2      | Most features by example . . . . .             | 5         |
| 2.3      | Encoding-dependent parameters . . . . .        | 8         |
| 2.4      | Advanced features . . . . .                    | 9         |
| <b>3</b> | <b>Configuration file</b>                      | <b>10</b> |
| <b>4</b> | <b>Kerning pairs that are often missing</b>    | <b>10</b> |
| 4.1      | Character combinations . . . . .               | 11        |
| 4.2      | Quotation marks . . . . .                      | 11        |
| <b>5</b> | <b>An example of how to optimize a font</b>    | <b>12</b> |
| <b>6</b> | <b>The implementation</b>                      | <b>16</b> |
| 6.1      | Class file . . . . .                           | 17        |
| 6.1.1    | Glyph classes . . . . .                        | 33        |
| 6.1.2    | Extra commands for special encodings . . . . . | 40        |
| 6.2      | Footer of <code>mtx</code> file . . . . .      | 40        |
| 6.3      | Class option files . . . . .                   | 41        |
| 6.3.1    | T1 encoding . . . . .                          | 41        |
| 6.3.2    | TS1 encoding . . . . .                         | 47        |
| 6.3.3    | OT1 encoding . . . . .                         | 52        |
| 6.3.4    | T2A encoding . . . . .                         | 58        |

|       |                        |    |
|-------|------------------------|----|
| 6.3.5 | T2A encoding . . . . . | 63 |
| 6.3.6 | LY1 encoding . . . . . | 68 |
| 6.4   | Templates . . . . .    | 72 |
| 6.4.1 | T1 encoding . . . . .  | 73 |
| 6.4.2 | TS1 encoding . . . . . | 80 |
| 6.4.3 | OT1 encoding . . . . . | 85 |

## Copyright

Copyright 2004 Harald Harders.

This program can be redistributed and/or modified under the terms of the LaTeX Project Public License Distributed from CTAN archives in directory macros/latex/base/lppl.txt; either version 1.3 of the License, or any later version.

## 1 Introduction

Every glyph of a font is surrounded by a bounding box. Have a look at these glyphs:



As can be seen in this example, the glyphs may extend the bounding box.

Normally, one character is placed after the other by simply putting the bounding boxes directly after each other:



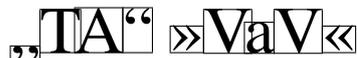
In most cases, this works great, but sometimes the distance between two glyphs is ugly then. Here, for instance, have a look at “VA”, “fT”, “VA”, and “fT”. To improve these cases, the bounding boxes are moved together or away from another, as shown in the next example. This is called kerning. To be able to do the kerning, the font contains a table of pairs of glyphs and the distance to move the second one.



What you see here is the original kerning of the Times Roman installed in your TeX system. The V–A pairs are improved; but the f still touches the T which is not wanted. The Times Roman font misses this kerning pair. Better were this:



Most fonts are missing many of these kerning pairs that are necessary for a good typography. Especially if you are writing in a language other than English, many kerning pairs are missing. Have a look at this example with quotation marks used in German, with the original kerning of the Times:



As you can see, there is no kerning at all.

And here comes, what this class is intended to do: It is difficult to calculate the necessary kerning automatically. Thus, it can be helpful to generate some kerning pairs and to try different amounts of kernings. Compare the different kernings in this example (this time without the bounding boxes):



The first one is without kerning. The kerning in the last one ( $-0.35$  em) is surely too much since both glyphs touch each other. It really isn't easy to find the "perfect" kerning. One remark here: It is worse to have a too tight kerning than too less kerning. Thus please do as little kerning as you think can work well.

With the `kerntest` class, it is easy to try out different values. How this can be done is described in the next section.

## 2 Usage of the class

### 2.1 Introduction

In the simplest case, you can use the package like this:

```
\documentclass[family=ptm,extraname=shortexample,footer=false]{kerntest}
\begin{document}
\begin{kerntable}
  \testkern{W}{-}{A}{-}{W} original \\
  \testkern{V}{-160}{A}{+160}{V} altered \\
\end{kerntable}
\end{document}
```

You have to define the font family to be tested in the optional argument of the `\documentclass` command. The syntax is `family=<font family>`, while *<font family>* is the typical abbreviation according to Karl Berry's name scheme [1], e.g., `cmr` for Computer Modern Roman, `ptm` for Times Roman, `phv` for Helvetica, `pmnj` for Adobe Minion with old-style numbers, `pmnx` for Adobe Minion with expert characters.

The options `extraname=shortexample` and `footer=false` are not so important and described later.

`kerntable`      The kerning table is generated by the `kerntable` environment which is represented by a `longtable` environment internally. Each line of this environment has  
`\testkern`      to contain one `\testkern` command with five arguments:

1. The first one contains the name or character number of the left character in the table (see Fig. 1). Which glyph corresponds to a given value depends on the used encoding, default is the cork encoding T1. The glyph can be

| Font t1-ptm-m-n-shortexample |      |      |      |      |             |             |             |              |          | 1 |
|------------------------------|------|------|------|------|-------------|-------------|-------------|--------------|----------|---|
| slot                         | name | orig | new  | both | k. 1        | k. 2        | orig.       | new          | comment  |   |
| 065                          | A    | WAW  | WAW  | WAW  | -120        | -90         | WAexampleAW | WAexampleAW  | original |   |
| 065                          | A    | VAV  | VA V | VAV  | (-135) -160 | (-135) +160 | VAexampleAV | VAexampleA V | altered  |   |
| slot                         | name | orig | new  | both | k. 1        | k. 2        | orig.       | new          | comment  |   |

Figure 1: Example for a part of a kerning table (explanations in the text)

specified by giving a decimal number (0 to 255), a hexadecimal number ("0 to "FF), an octal number ('0 to '377), or by giving the PostScript glyph name, e.g., `grave`, `guillemotright`, `A`, `Aring`. It does not work to give L<sup>A</sup>T<sub>E</sub>X sequences as `<<`, `\guillemotright`, etc.

2. The second argument gives the kerning of the characters defined in the first and in the third argument. The used unit are Postscript Type1 font units which have the length of 0.001 em. It is not allowed to specify another unit. If the second argument is "-" the original kerning of the font is shown (first line of the example). If a value is given (second line) the original kerning is overwritten by the given value. Negative values reduce the distance of the glyphs, positive values increase it.
3. The third argument specifies the second glyph.
4. The fourth argument is the kerning between the second and the third glyph and works exactly as the second argument.
5. The fifth argument specifies the third glyph.

After the `\testkern` command, an arbitrary (but short) comment may be added. Often, it is good to write the name of the glyph here. With `t1-XXX-m-n.tex` and `ts1-XXX-m-n.tex`, two templates are given that contains all glyph names for the T1 and TS1 encodings.

Each line in the `kerntable` environment (even the last one) has to end with a `\\` or `\tabularnewline`.

The output of this file is shown in Fig. 1. In the first column, the number of the middle character (argument #3 of the `\testkern` command) is listed, followed by the PostScript glyph name. In the third column, the combination of characters is printed with original kerning for both pairs, while the fourth column shows the newly suggested kerning. In the fifth column, both variants are printed over each other (the old one grey, the new one black). The next two columns show the values of the two kerning pairs in Postscript font units (normally, 0.001 em). If the user has not given a new kerning the original value is printed in grey. If the user has defined a new kerning, this new value is printed in black. If, in this case, the old kerning is unequal zero, it is printed in parentheses before the new value (second line of the example). The rest of each line are examples and comments.

## 2.2 Most features by example

The next example shows some more switches that can be defined by the user:

```
\listfiles
\documentclass[family=ptm]{kerntest}
\kernsetup{encoding=T1,series=bx,shape=n,example=M}
\kernsetup{size=17.28pt,baselineskip=17pt,papersize=a4paper}
\kernsetup{extraname=example,color=true,footer=false}

\newglyphclass{right}{A}{A,Aring,Adieresis,Abreve[500]}
\newglyphclass{left}{A}{A,Aring,Adieresis,Abreve[500]}
\newglyphclass{right}{fullstop}{period,comma}
\newglyphclass{left}{fullstop}{period,comma}

\begin{document}

\begin{kerntable}
\testkern{016}{-30}{046}{-30}{017} decimal \\
\testkern{"10}{-}{"2C}{-}{"11} hexadecimal \\
\testkern{'020}{-}{'101}{-80}'021} octal \\
\testkern{quotedblleft}{-}{Aring}{-80}{quotedblright} by name \\
\testkern{quotedblleft}{-100}{AE}{-}{quotedblright} \\
\testkern{quotedblleft}{-}{B}{-60}{quotedblright} \\
\testkern{quotedblleft}{-}{C}{-}{quotedblright} \\
\testkern{T}{-}{f}{+90}{T} \\
\testkern{quotedblbase}{-60}{T}{-}{quotedblleft} \\
\testkern{quotedblbase}{-}{Adieresis}{-}{quotedblleft} \\
\testkern{quotedblbase}{-}{A}{-200}{quotedblleft} \\
\testkern{quotedblbase}{-}{Aring}{-}{quotedblleft} \\
\testkern{quotedblbase}{-}{Abreve}{-}{quotedblleft} \\
\testkern{guillemotright}{-55}{V}{-55}{guillemotleft} \\
\end{kerntable}

\end{document}
```

Have a look at the results in Fig 2 before the switches are explained.

**\kernsetup** All class options except `family` can either be given as class option in the optional argument of the `\documentclass` command or as argument of the `\kernsetup` command. The `family` class option has to be given in the `\documentclass` command. Here comes a list of all class options:

- encoding** `encoding=<font encoding>`: Font encoding (default: T1). Currently, OT1, T1, TS1, T2A, T2B, and LY1<sup>1</sup> are supported.
- family** `family=<font family>`: Abbreviation of the font-family name according to Karl Berry's scheme [1]. This option is mandatory in the optional argument of the `\documentclass` command.
- series** `series=<font series>`: Abbreviation for the series of the font (default: m), e.g., m for medium, sb for semibold, b for bold, bx for bold extended.

<sup>1</sup>While the other encodings are generated starting from `.etx` files, the LY1 encoding has been extracted from `texnansi.enc`. Some glyphs may have incorrect names.

| Font t1-ptm-bx-n-example |           |      |     |      |      |           |       |       |             | 1 |
|--------------------------|-----------|------|-----|------|------|-----------|-------|-------|-------------|---|
| slot                     | name      | orig | new | both | k. 1 | k. 2      | orig. | new   | comment     |   |
| 046                      | period    | “.”  | “.” | “.”  | -30  | (-55)-30  | “.M.” | “.M.” | decimal     |   |
| 044                      | comma     | “,”  | “,” | “,”  | -30* | (-45)-30* | “,M,” | “,M,” | hexadecimal |   |
| 065                      | A         | “A”  | “A” | “A”  | -10  | -80       | “AMA” | “AMA” | octal       |   |
| 197                      | Aring     | “Å”  | “Å” | “Å”  | -10  | -80†      | “ÅMÅ” | “ÅMÅ” | by name     |   |
| 198                      | AE        | “Æ”  | “Æ” | “Æ”  | -100 | 0         | “ÆMÆ” | “ÆMÆ” |             |   |
| 066                      | B         | “B”  | “B” | “B”  | 0    | -60       | “BMB” | “BMB” |             |   |
| 067                      | C         | “C”  | “C” | “C”  | 0    | 0         | “CMC” | “CMC” |             |   |
| 102                      | f         | TfT  | TfT | TfT  | 0    | +90       | TfMfT | TfMfT |             |   |
| 084                      | T         | „T“  | „T“ | „T“  | -60  | 0         | „TMT“ | „TMT“ |             |   |
| 196                      | Adieresis | „Ä“  | „Ä“ | „Ä“  | 0    | 0         | „ÄMÄ“ | „ÄMÄ“ |             |   |
| 065                      | A         | „A“  | „A“ | „A“  | 0    | -200      | „AMA“ | „AMA“ |             |   |
| 197                      | Aring     | „Å“  | „Å“ | „Å“  | 0    | -200*     | „ÅMÅ“ | „ÅMÅ“ |             |   |
| 128                      | Abreve    | „Ă“  | „Ă“ | „Ă“  | 0    | -100*     | „ĂMĂ“ | „ĂMĂ“ |             |   |
| 086                      | V         | »V«  | »V« | »V«  | -55  | -55       | »VMV« | »VMV« |             |   |
| slot                     | name      | orig | new | both | k. 1 | k. 2      | orig. | new   | comment     |   |

Figure 2: Example for a part of a kerning table (explanations in the text). The labels marked with a star are described in Section 2.4.

- shape** `shape=(font shape)`: Abbreviation for the font shape (default: `n`), e.g., `n` for up-right, `it` for italic, `sl` for slanted, `sc` for small caps, `scit` for italic small caps.
- size** `size=(font size)`: Size of the tested font (default: 17.28pt) in arbitrary units. This sets the `\baselineskip` to 1.2 times the given value. The `size` option does not change the size of the legend text which is fixed to 10 pt.
- baselineskip** `baselineskip=(baselineskip)`: Sets the `\baselineskip` explicitly. To take effect, it has to be given after the option `size` (default: `1.2*17.28pt`).
- designsize** `designsize=(design size)`: For calculating the kerning data, a PostScript font unit is used which is 1/1000 of the font’s design size. Unfortunately, it is not possible to get this size reliably in L<sup>A</sup>T<sub>E</sub>X. For most fonts, 1 em corresponds to the design size. But in some cases, it is not true:

1. Some fonts have a different em length, for example, the Computer Modern fonts. Then, the size given by the option `size` corresponds to the design size, but 1 em does not.
2. If the font is scaled by the `.fd` file, 1 em is also scaled and may correspond to the design size while the size given with `size` does not.

Since in most cases, 1 em is the correct design size, this is the default. If it is not, you can give the correct design size by using the option `designsize`, for example, `designsize=0.9em`, `designsize=17pt`.

- example** `example=<text>`: Alters the example text for columns 5 and 6 (default: `example`).
- papersize** `papersize=<papersize>` Tells the `geometry` package which pagesize to use. Supported are all pagesizes handled by `geometry`, e.g., `a4paper`, `letter`, `legal` (no default).
- extraname** `extraname=<fontname extension>`: Normally, the heading of each page of output as well as the filename of the `mtx` file are generated automatically by appending encoding, font family, font series, and font shape, e.g., `t1-cmr-m-n.mtx`. If you use this option, `-<fontname extension>` is added both to the headings and to the filename. For example, `extraname=test1` leads to `t1-cmr-m-n-test1.mtx`. This is useful if you want to generate different `mtx` files that normally got the same name.<sup>2</sup>
- color** `color=<true/false>`: Switches on color output (default: `false`). New values are printed in red, while the old ones are printed black instead of black/grey.
- copyquotation** `copyquotation=<true/false>`: If a kerning pair containing a double quotation mark, including guillemots, is set, write also the corresponding single one to the `mtx` file.
- writeall** `writeall=<true/false>`: Write also the original kerning data to the `mtx` file.
- footer** `footer=<true/false>`: Switch on or off the footline.<sup>3</sup>

It has been mentioned some times that an `mtx` file is generated automatically. `mtx` files contain the font metrics during the `fontinst` process. Amongst other things, they contain the kerning data. For example, the `mtx` file generated by the last example looks like this:

```
%%
%% This is file 't1-ptm-bx-n-example.mtx',
%% generated on 2004/4/14 by kerntest.cls, (c) 2004 Harald Harders.
%%
%% The original source file was:
%%
%% t1-ptm-bx-n-example (.tex?) with these font options:
%% Encoding: T1
%% Family:   ptm
%% Series:   bx
%% Shape:    n
%% User-defined name: -example
%%
\relax
\metrics
\needsfontinstversion{1.926}
%%
%% Kerning data for single characters and
%% the first members of the glyph classes.
%%
```

<sup>2</sup>This is why the first example used this option.

<sup>3</sup>This option seems only to work in the `\documentclass` options. Don't ask me why.

```

%% After each \setkern entry, the glyph classes
%% for both glyphs are given (./. means no class).
%%
\setkern{quotedblleft}{period}{-30}% ./ -- left/fullstop
\setkern{period}{quotedblright}{-30}% right/fullstop -- ./
\setkern{A}{quotedblright}{-80}% right/A -- ./
\setkern{quotedblleft}{AE}{-100}% ./ -- ./
\setkern{B}{quotedblright}{-60}% ./ -- ./
\setkern{f}{T}{+90}% ./ -- ./
\setkern{quotedblbase}{T}{-60}% ./ -- ./
\setkern{A}{quotedblleft}{-200}% right/A -- ./
\setkern{guillemotright}{V}{-55}% ./ -- ./
\setkern{V}{guillemotleft}{-55}% ./ -- ./
%%
%% Kerning factors for the different glyph classes.
%%
\setleftkerning{Aring}{A}{1000}% left/A
\setleftkerning{Adieresis}{A}{1000}% left/A
\setleftkerning{Abreve}{A}{500}% left/A
\setleftkerning{comma}{period}{1000}% left/fullstop
%%
\setrightkerning{Aring}{A}{1000}% right/A
\setrightkerning{Adieresis}{A}{1000}% right/A
\setrightkerning{Abreve}{A}{500}% right/A
\setrightkerning{comma}{period}{1000}% right/fullstop
%%
\endmetrics
%%
%% End of file 't1-ptm-bx-n-example.mtx'.

```

Only new or changed kerning values are inserted (e.g., `quotedblleft-A` is not included).

`\mtxcomment` Using the `\mtxcomment{<comment>}` command, you can write the given argument as comment into the `mtx` file.

## 2.3 Encoding-dependent parameters

Some encodings may have slight differences depending on the used shape. For example, typewriter fonts may have ligatures but they are normally not used. Thus, the encodings do not have some glyphs when used with typewriter fonts (e.g., `ff`, `fi`, `ffi`, `fl`, `fl` are missing).

`\encodingsetup` The class provides an interface to give the necessary parameters to these encodings. Use the command `\encodingsetup` that takes a comma-separated list of options as argument (as `\kernsetup`).

Here are the encoding-specific options:

### T1 encoding:

`ligaturing` `ligaturing=<number>`: Level of how many ligatures are used (-2, -1, 0, or 1, default 1). Here is the description from `t1.etx` (version 1.923, 2002/10/29):

- 1 All the standard ligature glyphs (`fi`, `fl`, `ff`, `ffi`, `ffl`, `IJ`, and `ij`) are included and the normal ligaturing instructions (those for the `f`-ligatures) are included.
- 0 All the standard ligature glyphs are included, but none of their ligaturing instructions.
- 1 The seven slots normally used for ligatures are left empty.
- 2 The seven slots normally used for ligatures are left empty, as are the slots normally used for `c`, `f`, `s`, `i`, and `I`.

### OT1 encoding:

|                          |  |
|--------------------------|--|
| <code>ligaturing</code>  | <code>ligaturing=&lt;number&gt;</code> : Level of how many ligatures are used (0, 1, or 2, default 2). Please refer to <code>ot1.etx</code> for more details.                  |
| <code>italicizing</code> | <code>italicizing=&lt;true/false&gt;</code> : Use <code>dollar</code> when <code>false</code> and <code>sterling</code> when <code>true</code> (default: <code>false</code> ). |

## 2.4 Advanced features

In most fonts, different glyphs need the same kerning because their left or right edges are very similar, for example, the kerning on the left sides of `B`, `D`, `Đ`, `Ð`, `E`, `Ě`, `Ě`, `É`, `Ê`, `Ë`, `F`, `H`, `I`, `İ`, `Í`, `Î`, `Ï`, `IJ`, `J`, `K`, `L`, `Ł`, `L`, `Ł`, `N`, `Ń`, `Ň`, `Ñ`, `P`, `R`, `Ř`, `Ŕ`, `Ŗ`, and `Þ` should be equal.

```
\defglyphclass
\newglyphclass
\renewglyphclass
\provideglyphclass
```

This can be reached by using so called “glyph classes”. A new glyph class can be defined using one of the commands `\defglyphclass`, `\newglyphclass`, `\renewglyphclass`, and `\provideglyphclass`. The differences are similar to these of the commands `\newcommand`, `\renewcommand`, etc. They all have the syntax `\defglyphclass{<side>}{<name>}{<glyphlist>}`. `<side>` specifies the side of the glyphs on which the kerning shall be equal (`left` or `right`). The parameter `<name>` specifies the name of the glyph list, the list above could be named “`H`” because they all have a similar shape as the `H`. The third argument, `<glyphlist>`, contains a comma-separated list of all glyphs (PostScript names or numbers—as usual). For example, the above list is build by this command:<sup>4</sup>

```
%\newglyphclass[left]{H}{B,D,Dcaron,Eth,E,Ecaron,Eogonek,Egrave,%
% Eacute,Ecircumflex,Edieresis,F,H,I,Idotaccent,Igrave,Iacute,%
% Icircumflex,Idieresis,IJ,J,K,L,Lacute,Lcaron,Lslash,N,Nacute,%
% Ncaron,Ntilde,P,R,Racute,Rcaron,Ng,Thorn}
```

You can specify arbitrary glyph classes. If you, for example, use the `copyquotation` option glyph classes are made containing one double and one single quotation mark each.

When you write a kerning table using the `kerntable` environment and it happens that you change the kerning for a glyph that is member of a glyph class, the kernings for all other glyphs of the same glyph class are automatically changed on the specific side. This can be seen in the example on Page 5 and in Figure 2:

`period` and `comma` build a glyph pair on both sides. In the first line of the table, left and right kerning between the period and the quotation marks are changed by the user. The kerning between the comma and quotation marks is then set

<sup>4</sup>Due to a problem between the `ltxdoc` and `verbatim` packages, the `%` signs appear at the beginning of each line. Just delete them in mind.

automatically; the user does not have to specify them again (the kerning data contain a simple - in the second line). If you specify the same value explicitly, a warning is generated. If you specify conflicting values, the programme generates an error message (not shown).

Automatically generated kerning pairs are marked by \* behind the value (as can be seen in the second and 12th line of the example). Repeated values are marked by †.

There is one shortcoming: If you don't specify the kerning for a glyph class at the first occurrence of this glyph, the correct kerning data are not shown for the occurrences before the position you have specified the kerning. In the example, the kerning between the members of the glyph class "A" (A, Ä, and Å) and the right German quotation mark ("), is not specified for the first char of the glyph class, Ä, but for the second one, A. Thus, the kerning of -150 is shown for "A" and for "Ä", but not for "Å". But nevertheless, the kerning data written to the mtX files are correct.

All glyphs, given by `\defglyphclass` get the same kerning by default. You can specify different scale factors by appending [*scale*] to each glyph name; while a factor of 1000 is the default and means "the same kerning width".

For example,

```
%\defglyphclass{right}{A}{A,Aring[800],Adieresis[1200],Abreve}
```

defines a glyph class containing "A", "Å", "Ä", and "Ǻ". All kernings on the right side of "Ǻ" have a width of 80 of these "A" has. "Ä" is kerned 120% of "A". "Ǻ" is again kerned as "A".

You can also specify a different scaling for the first glyph in the glyph class. But then, all values are scaled in order to reach a factor of 1000 for the first entry. For example,

```
%\defglyphclass{right}{A}{A[500],Aring[400],Adieresis[600],Abreve[500]}
```

is identical to the example above.

The effect of scaled kernings can be seen in the example on Page 5 and in Figure 2 where "Ǻ" has half the scaling of "A" and "Å".

If two glyphs with scalings different from 1000 meet each other both scaling factors are multiplied.

There are some interesting commands to handle these glyph classes. Please have a look in Section 6.1.1 for their description.

### 3 Configuration file

If you are too lazy to put the same options into every source file you may write all options except `family` into the configuration file `kerntest.cfg` and put it into the L<sup>A</sup>T<sub>E</sub>X path. If it is present, it is loaded automatically.

### 4 Kerning pairs that are often missing

This section shows some kerning pairs that are often missing, even in expensive fonts. This problem arises since most fonts are merely designed for the English language.

## 4.1 Character combinations

Some glyphs need a kerning to many other glyphs, including “A”, “T”, “V”, “W”, and “Y”. For the ordinary lowercase letters, these kernings are included in most fonts (if the lowercase letter is on the right of the capital). But often, glyphs of other Languages than English are forgotten, e.g., “Tç”, “Vé”, etc. But you may not simply copy all kernings of, for instance, “Ta” to “Tä”, “Tá”, “Tâ”, “Tã” etc. Often, these glyphs have parts that force the kerning to be reduced or even deleted.

Most character pairs with the uppercase letters (“A”, “T”, “V”, “W”, “Y”) after a lowercase letters are not kerned in the fonts. In most cases, this should not be a problem because these combinations are never printed. (Nowadays, its getting more and more important to have these kernings since it is a fashion to use uppercase letters within words, e.g., “ServicePoint”.<sup>5</sup>) But some combinations really should be kerned: “eV” (electronvolt), “mV” (millivolt).

## 4.2 Quotation marks

Most fonts don’t provide kerning for quotation marks other than the English ones. In English, “Hello” is used (“66–99”). French uses « and »: «Bonjour». In German, the three possibilities „Hallo“ (“99–66”), »Hallo«, and »Hallo« are used. In Italian, «Ciao» or “Ciao,, are possible. Swedish uses »Hi» or ”Hi” [2]. For all non-English possibilities, most fonts have no kerning information. Thus, you should generate five tables for every font, hopefully containing all possibilities («H», »H«, “H”, „H“, ”H,): `\testkern{019}{-}{\langle glyph \rangle}{-}{020}`, `\testkern{020}{-}{\langle glyph \rangle}{-}{019}`, `\testkern{016}{-}{\langle glyph \rangle}{-}{017}`, `\testkern{018}{-}{\langle glyph \rangle}{-}{016}`, `\testkern{017}{-}{\langle glyph \rangle}{-}{018}`, where `\langle glyph \rangle` stands for all 256 glyphs contained in a T1 encoded font.

All problems mentioned for the double quotation marks apply also for their single variants („, ‘, ’, ‹, and ›). In most cases, they need the same kerning as their double counterparts.

There are two templates enclosed to build these kerning tables: `t1-XXX-m-n.tex` and `ts1-XXX-m-n.tex`. Hopefully, the names tell you which one to use. They contain some comments that should help to use them.

One (repeated) remark: Please don’t overdo the kerning if you adjust it. In most cases, good is less than you think on first sight. You can also orientate on predefined kernings. For example, the kerning for A“ should be similar to A”.

But you are still not save to get the correct kerning when your font knows them. This is due to the fact that there are multiple possibilities to access the quotation marks. For example, « can be produced by `<<`, `\guillemotleft`, and even by `\symbol{19}` (if you are using T1 encoding). If you use `inputenc.sty` you may use the characters directly, e.g., `<`. And after loading `babel.sty`, you can use `\flqq` and when writing German “<.

These possibilities are not equivalent.

The direct commands `\textquotedblleft`, `\textquotedblright`, `\quotedblbase`, `\guillemotleft`, `\guillemotright`, `\textquoteleft`, `\textquoteright`, `\quotesinglbase`, `\guilsinglleft`, and `\guilsinglright` work properly; they kern on their left and their right side.

The directly written quotation marks («, », etc.) also work correctly because the corresponding encoding file (e.g., `latin1.def`) translates them to the direct

---

<sup>5</sup>The German railway company really uses this term!

commands.

The ligatures ‘ ‘ and ’ ’ seem also to work correctly. But the ligature , , kerns correctly on its right side, but on its left side, it kerns as a comma. This may also be correct but it needn't be always the case. << and >> do not kern at all on their left side.<sup>6</sup>

Looking at the `babel` commands, only `\grqq` and `\grq` surely work correct. The others (`\glqq`, `\glq`, `\flqq`, `\flq`, `\frqq`, `\frq`) are defined differently and thus do not guarantee to kern correctly. On 2003/04/01, I have posted a bug report. Let's see what happens.

The `babel` shortcuts "‘, "’, "<, and "> work as good as the corresponding commands.

If you want a correct behaviour of all `babel` quotation marks, just copy the definition of `\grqq` (it contains of three command definitions!) from `babel.def` into your code and change it according to produce the other quotation marks.

## 5 An example of how to optimize a font

In this section, a very simple example is shown how to install a single font shape with `fontinst` [4] and how to change kernings for it. If you really want to understand what happens read the `fontinst` manual [4], “`TeX Unbound`” by Alan Hoenig [3], or “`The Font Installation Guide`” by Philipp Lehman [5].

Ghostscript contains the font “`Century Schoolbook L Roman`” which is shipped as files `c0590131.afm` and `c0590131.pfb`. Please copy these files into a temporary directory.

According to Karl Berry's scheme, the fontname is `uncr8a`. But this font is already prepared on most `TeX` systems. Thus we take the fontname `9ncr8a` here. This will be the name for the result file.

Then, run `TeX` (not `LATeX`) on the script `schoolb1.tex` which does most work to install a new font:

```
\input fontinst.sty
\nneedsfontinstversion{1.914}
% input AFMs:
\transformfont{9ncr8r}{\reencodefont{8r}{\fromafm{c0590131}}}
\fromafm{9ncr8r}
% install fonts:
\installfonts
% declare the font familys for T1 and TS1 encoding:
\installfamily{T1}{9nc}{}
\installfamily{TS1}{9nc}{}
% install a raw font:
\installrawfont{9ncr8r}{9ncr8r,8r}{8r}{}{}{}{}{}
% install the fonts in T1 and TS1 encoding:
\installfont{9ncr8t}{9ncr8r,latin}{T1}{T1}{9nc}{m}{n}{}
\installfont{9ncr8c}{9ncr8r,textcomp}{TS1}{TS1}{9nc}{m}{n}{}
% ready:
\endinstallfonts
```

---

<sup>6</sup>I believe they kern as < reps. >. But these characters don't have any kerning information in most cases.

## Some problematic kernings

»V«, “A”, „VA“, „VĀ“  
>V<, ‘A’, ,VA‘, ,VĀ‘

Figure 3: Font example for Century Schoolbook L with original kerning

`\bye`

This run creates some files with the extensions `.pl` and `.vpl`. They have to be converted to `tfm` and `vf` files as follows:

```
pltotf 9ncr8r.pl
vptovf 9ncr8c.vpl
vptovf 9ncr8t.vpl
```

Now, you can delete the temporary files with the extensions `.mtx`, `.pl`, and `.vpl`.

The new font is ready for use with  $\text{\LaTeX}$ , now (only for T1 and TS1 encoding, OT1 encoding has been left out). Just run  $\text{\LaTeX}$  on the test file `testschoolb.tex`. But you are not yet able to use `dvips` or `PDF $\text{\LaTeX}$`  because they need a `map` file. The corresponding one, `schoolb.map` looks like this:

```
9ncr8r CenturySchL-Roma "TeXBase1Encoding ReEncodeFont" <8r.enc <c0590131.pfb
```

With help of this map file, the `dvi` file can be converted to Postscript using `dvips`: `dvips -u +./schoolb.map -o testeschoolb-1.ps testschoolb`. Unfortunately, you cannot use `PDF $\text{\LaTeX}$`  without adding the contents of `schoolb.map` to the global `map` file.

When viewing the result in `testschoolb-1.ps` (Fig. 3), you see that this specific font already has most kernings that are missing in other fonts. The only really forgotten kernings are `A“`, `A‘`, `Ā“`, and `,V`.

Imagine that many kernings were unsatisfactory. Then, we generate a kerning table containing the glyph combinations we do not like:

```
\listfiles
\documentclass[family=9nc,footer=false]{kerntest}
\kernsetup{encoding=T1,series=m,shape=n,example=tst,extraname=1}
\kernsetup{size=14.40pt,baselineskip=16.5pt,papersize=a4paper}
\renewcommand\thepage{}
\newglyphclass{left}{A}{A,Abreve[500]}
\newglyphclass{right}{A}{A,Abreve[500]}
\begin{document}
\begin{kerntable}
\testkern{020}{-200}{086}{-200}{019}  \ \
\testkern{016}{-220}{065}{-220}{017}  \ \
\testkern{018}{-}{065}{-220}{016}  \ \
\testkern{018}{-220}{086}{-}{016}  \ \
\testkern{015}{-200}{086}{-200}{014}  \ \
\testkern{096}{-220}{065}{-220}{039}  \ \
\testkern{013}{-}{065}{-220}{096}  \ \
```

| Font t1-9nc-m-n-1 |      |      |     |      |             |             |         |         |         |
|-------------------|------|------|-----|------|-------------|-------------|---------|---------|---------|
| slot              | name | orig | new | both | k. 1        | k. 2        | orig.   | new     | comment |
| 086               | V    | »V«  | »V« | »V«  | (-85)-200   | (-85)-200   | »VtstV« | »VtstV« |         |
| 065               | A    | “A”  | “A” | “A”  | (-65)-220   | (-67)-220   | “AtstA” | “AtstA” |         |
| 065               | A    | „A“  | „A“ | „A“  | +36         | -220        | „AtstA“ | „AtstA“ |         |
| 086               | V    | „V“  | „V“ | „V“  | (-81)-220   | 0           | „VtstV“ | „VtstV“ |         |
| 086               | V    | ›V‹  | ›V‹ | ›V‹  | (-85)-200   | (-85)-200   | ›VtstV‹ | ›VtstV‹ |         |
| 065               | A    | ‘A’  | ‘A’ | ‘A’  | (-65)-220   | (-66)-220   | ‘AtstA’ | ‘AtstA’ |         |
| 065               | A    | ,A‘  | ,A‘ | ,A‘  | 0           | -220        | ,AtstA‘ | ,AtstA‘ |         |
| 086               | V    | ,V‘  | ,V‘ | ,V‘  | -220        | 0           | ,VtstV‘ | ,VtstV‘ |         |
| 086               | V    | AVA  | AVA | AVA  | (-101)-220  | (-100)-220  | AVtstVA | AVtstVA |         |
| 086               | V    | ǺVǺ  | ǺVǺ | ǺVǺ  | (-101)-110* | (-100)-110* | ǺVtstVǺ | ǺVtstVǺ |         |
| slot              | name | orig | new | both | k. 1        | k. 2        | orig.   | new     | comment |

Figure 4: Kerning table for Century Schoolbook L

```

\testkern{013}{-220}{086}{-}{096} \\
\testkern{065}{-220}{086}{-220}{065} \\
\testkern{Abreve}{-}{086}{-}{Abreve} \\
\end{kerntable}
\end{document}

```

This leads to the output shown in Fig. 4 and to the mtX file t1-9nc-m-n-1.mtx:

```

%%
%% This is file 't1-9nc-m-n-1.mtx',
%% generated on 2004/4/14 by kerntest.cls, (c) 2004 Harald Harders.
%%
%% The original source file was:
%%
%% t1-9nc-m-n-1 (.tex?) with these font options:
%% Encoding: T1
%% Family: 9nc
%% Series: m
%% Shape: n
%% User-defined name: -1
%%
\relax
\metrics
\needsfontinstversion{1.926}
%%
%% Kerning data for single characters and
%% the first members of the glyph classes.
%%
%% After each \setkern entry, the glyph classes
%% for both glyphs are given (./. means no class).
%%

```

```

\setkern{guillemotright}{V}{-200}% ./ -- ./
\setkern{V}{guillemotleft}{-200}% ./ -- ./
\setkern{quotedblleft}{A}{-220}% ./ -- left/A
\setkern{A}{quotedblright}{-220}% right/A -- ./
\setkern{A}{quotedblleft}{-220}% right/A -- ./
\setkern{quotedblbase}{V}{-220}% ./ -- ./
\setkern{guilsinglright}{V}{-200}% ./ -- ./
\setkern{V}{guilsinglleft}{-200}% ./ -- ./
\setkern{quoteleft}{A}{-220}% ./ -- left/A
\setkern{A}{quoteright}{-220}% right/A -- ./
\setkern{A}{quoteleft}{-220}% right/A -- ./
\setkern{quotesinglbase}{V}{-220}% ./ -- ./
\setkern{A}{V}{-220}% right/A -- ./
\setkern{V}{A}{-220}% ./ -- left/A
%%
%% Kerning factors for the different glyph classes.
%%
\setleftkerning{Abreve}{A}{500}% left/A
%%
\setrightkerning{Abreve}{A}{500}% right/A
%%
\endmetrics
%%
%% End of file 't1-9nc-m-n-1.mtx'.

```

Using this file, you can repeat the fontinst run with a slightly changed script schoolb2.tex:

```

\input fontinst.sty
\needsfontinstversion{1.914}
% input AFMs:
\transformfont{9ncr8r}{\reencodefont{8r}{\fromafm{c0590131}}}
\fromafm{9ncr8r}
% install fonts:
\installfonts
% declare the font familys for T1 and TS1 encoding:
\installfamily{T1}{9nc}{}
\installfamily{TS1}{9nc}{}
% install a raw font:
\installrawfont{9ncr8r}{9ncr8r,8r}{8r}{}{}{}{}
% install the fonts in T1 and TS1 encoding:
\installfont{9ncr8t}{t1-9nc-m-n-1,9ncr8r,latin}{T1}{T1}{9nc}{m}{n}{}
\installfont{9ncr8c}{9ncr8r,textcomp}{TS1}{TS1}{9nc}{m}{n}{}
% ready:
\endinstallfonts
\bye

```

The only change amongst schoolb1.tex is the added entry t1-9nc-m-n-1, at the beginning of the second argument of the `\installfont{9ncr8t}` command. This includes the new kernings into the generated font. After repeating also the `pltotf` and `vptovf` calls, you can use the font with the new kernings. Running

### Some problematic kernings

»V«, “A”, „VA“, „VĂ“  
›V‹, ‘A’, ,VA‘, ,VĂ‘

### Some problematic kernings

»V«, “A”, „VA“, „VĂ“  
›V‹, ‘A’, ,VA‘, ,VĂ‘

Figure 5: Font example for Century Schoolbook L with original (top) and modified (bottom) kerning. The kerning is much too strong. Here, it only shows the effect of altering the kerning.

$\LaTeX$  and `dvips` again on `testschoolb.tex` gives the output of Fig. 5. Here, the kerning values are much too strong. The only aim of this was to show a clear difference between original and modified kerning. Have a look at the  $\check{A}$  kernings. They have been set to be half as large as the A kernings on both sides.

The last thing to do is to install the font files into the corresponding paths of your  $\TeX$  distribution and to append the map information to the global map files (normally by using `updmap`).

## References

- [1] Karl Berry. Fontname, May 2003. <ftp://ftp.dante.de/tex-archive/info/fontname/>.
- [2] Friedrich Forssman, Ralf de Jong. Detailtypografie, Verlag Hermann Schmidt, Mainz, Germany, 2002.
- [3] Alan Hoenig.  $\TeX$  Unbound— $\LaTeX$  &  $\TeX$  Strategies for Fonts, Graphics, & More, Oxford University Press, 1998.
- [4] Alan Jeffry, Rowland McDonnell. `fontinst`—Font installation software for  $\TeX$ , June 1998. <ftp://ftp.dante.de/tex-archive/fonts/utilities/fontinst/>.
- [5] Philipp Lehman. The Font Installation Guide, August 2003. <ftp://ftp.dante.de/tex-archive/info/Type1fonts/fontinstallationguide.pdf>.

## 6 The implementation

Heading of all files.

```
1 <class>\ProvidesClass{kerntest}
2 <mtx & t1>\ProvidesFile{t1mtx.clo}
3 <mtx & ts1>\ProvidesFile{ts1mtx.clo}
4 <mtx & ot1>\ProvidesFile{ot1mtx.clo}
```

```

5 <mtx & t2a>\ProvidesFile{t2amtx.clo}
6 <mtx & t2b>\ProvidesFile{t2bmtx.clo}
7 <mtx & ly1>\ProvidesFile{ly1mtx.clo}
8 <version>\ProvidesFile{krntst-v.tex}
9 <class | mtx | version> [2004/04/14 v1.32 Generate kerning tables]

```

## 6.1 Class file

Use a standard class as base.

```

10 <*class>
11 \LoadClass[10pt]{article}

```

Use most of the space on the paper.

```

12 \RequirePackage[top=18mm,left=15mm,right=15mm,bottom=20mm]{geometry}

```

Font for the legends.

```

13 \renewcommand*\familydefault{\sfdefault}
14 \RequirePackage{helvet}

```

More required packages.

```

15 \RequirePackage{calc}
16 \RequirePackage{longtable}
17 \RequirePackage{array}
18 \RequirePackage{color}
19 \RequirePackage{ifthen}
20 \RequirePackage{keyval}

```

Layout settings.

```

21 \pagestyle{myheadings}
22 \def\@oddfoot{Kerning data, marked with $\ast$, are automatically reused
23   from values given before.
24   Repeated values are marked by $\dagger$.\hfill}
25 \def\@evenfoot{\@oddfoot}
26 \setlength{\parindent}{0mm}

```

Declare lengths for the font size and the baselineskip.

```

27 \newlength\krntst@size
28 \newlength\krntst@baselineskip

```

Set the default values for the class options.

```

29 \def\krntst@encoding{T1}
30 \def\krntst@series{m}
31 \def\krntst@shape{n}
32 \setlength\krntst@size{17.28pt}
33 \setlength\krntst@baselineskip{1.2\krntst@size}
34 \def\krntst@example{example}
35 \def\krntst@extraname{}
36 \definecolor{oldcolor}{gray}{0.5}
37 \definecolor{newcolor}{gray}{0}
38 \newboolean{krntst@writeall}

```

The design size is given as command rather than as length because it shall not be calculated to a real length (in pt), but it shall scale with the chosen font.

```

39 \newcommand\krntst@designsize{1em}

```

Process the class options using the keyval package.

```
40 \def\ProcessOptionsWithKV#1{%
41   \let\@tempc\relax
42   \let\KV@tempa\@empty
43   \edef\KV@tempa{%
44     \noexpand\setkeys{#1}{%
45       \optionlist{\@currname.\@currentx}%
46     }%
47   }%
48   \KV@tempa
49   \let\CurrentOption\@empty
50 }
```

Define the keys for the class options and the `\kernsetup` command.

```
51 \define@key{krntst}{encoding}{\def\krntst@encoding{#1}}
52 \define@key{krntst}{family}{\def\krntst@family{#1}}
53 \define@key{krntst}{series}{\def\krntst@series{#1}}
54 \define@key{krntst}{shape}{\def\krntst@shape{#1}}
55 \define@key{krntst}{size}{%
56   \setlength\krntst@size{#1}%
57   \setlength\krntst@baselineskip{1.2\krntst@size}%
58 }
59 \define@key{krntst}{baselineskip}{\setlength\krntst@baselineskip{#1}}
60 \define@key{krntst}{designsize}{\def\krntst@designsize{#1}}%
61 \define@key{krntst}{example}{\def\krntst@example{#1}}
62 \define@key{krntst}{papersize}{\geometry{#1}}
63 \define@key{krntst}{extraname}{\def\krntst@extraname{-#1}}
64 \define@key{krntst}{color}[true]{%
65   \csname if#1\endcsname
66   \definecolor{oldcolor}{gray}{0}%
67   \definecolor{newcolor}{rgb}{1,0,0}%
68   \else
69   \definecolor{oldcolor}{gray}{0.5}%
70   \definecolor{newcolor}{gray}{0}%
71   \fi
72 }
```

Do the copying of quotation marks by introducing glyph classes.

```
73 \define@key{krntst}{copyquotation}[true]{%
74   \csname if#1\endcsname
75   \newglyphclass{left}{leftguillemots}{guillemotleft,guilsinglleft}%
76   \newglyphclass{right}{leftguillemots}{guillemotleft,guilsinglleft}%
77   \newglyphclass{left}{rightguillemots}{guillemotright,guilsinglright}%
78   \newglyphclass{right}{rightguillemots}{guillemotright,guilsinglright}%
79   \newglyphclass{left}{leftquotes}{quotedblleft,quoteleft}%
80   \newglyphclass{right}{leftquotes}{quotedblleft,quoteleft}%
81   \newglyphclass{left}{rightquotes}{quotedblright,quoteright}%
82   \newglyphclass{right}{rightquotes}{quotedblright,quoteright}%
83   \newglyphclass{left}{basequotes}{quotedblbase,quotesinglbase}%
84   \newglyphclass{right}{basequotes}{quotedblbase,quotesinglbase}%
85   \fi
86 }
87 \define@key{krntst}{writeall}[true]{%
88   \setboolean{krntst@writeall}{#1}%
89   \ClassWarningNoLine{kerntest}{You are writing the new and the
```

```

90     original kerning data\MessageBreak
91     to the mtx file (option 'writeall'). Normally, it is\MessageBreak
92     not necessary to write original data}%
93 }
94 \define@key{krntst}{footer}[true]{%
95   \csname if#1\endcsname
96   \else
97     \def\@oddfoot{}%
98     \def\@evenfoot{\@oddfoot}%
99   \fi
100 }

```

`\kernsetup` Define the macro `\kernsetup` and make it available only in the preamble.

```

101 \newcommand\kernsetup{\setkeys{krntst}}
102 \@onlypreamble\kernsetup

```

Read in the configuration file if available. Do it before processing the options to allow the options to overwrite the configuration file entries.

```

103 \AtEndOfClass{%
104   \InputIfFileExists{kerntest.cfg}{%
105     \message{Configuration file 'kerntest.cfg' loaded.}%
106   }{%
107     \message{No configuration file 'kerntest.cfg' found.}%
108   }

```

Now, process the class options.

```

109 \ProcessOptionsWithKV{krntst}

```

This has to do something with a problem in `keyval.sty`. I do not really know what it does exactly.

```

110 \let\@unprocessedoptions\relax
111 }

```

Generate an error message if the class option family has not been given in the `\documentclass` command.

```

112 \ifx\krntst@family\relax
113   \ClassError{kerntest}{Class option family not or incorrect
114     given\@gobble}{%
115     You have to specify the font family by using the
116     class\MessageBreak
117     option family=<fontfamily>}%
118   \stop
119 \fi

```

Redefine the family option to be unusable in the `\kernsetup` command.

```

120 \AtEndOfClass{%
121   \define@key{krntst}{family}{%
122     \ClassError{kerntest}{Option 'family' used outside
123       \string\documentclass\space command}{%
124       The option 'family=<fontfamily>' has to be specified in the
125       optional argument\MessageBreak
126       of the \string\documentclass\space command.}%
127   }
128 }

```

```

\mtxcomment Define a command that writes a comment to the mtx file.
129 \newcommand\mtxcomment[1]{%
130   \protected@write\mtxfile{}\@percentchar\space #1}%
131 }

Define a command that is used to access the font for the legends.
132 \newcommand\krntst@helpfont{\normalfont\normalsize}

An internal counter that stores the slot of a glyph.
133 \newcounter{@glyphslot}%

The following commands have to be done at \begin{document} to ensure that all
\kernsetup calls have been made before.
134 \AtBeginDocument{%

Load all used encodings and T1 for the legends. If T1 is used, it is loaded twice; it
does not seem to be bad.
135   \RequirePackage[\krntst@encoding,T1]{fontenc}

Load the file that provides the Postscript glyph names. The trick to make it
lowercase ist stolen from the fontenc package.
136   \edef\reserved@f{%
137     \lowercase{\def\noexpand\reserved@f{\krntst@encoding mtx.clo}}}%
138   \reserved@f
139   \InputIfFileExists\reserved@f{%
140     \ClassWarningNoLine{kerntest}{Postscript name file '\reserved@f'
141       not found.\MessageBreak
142       The kerning table will be okay, but the generated mtx file will
143       not be usable}%
144   \newcommand\getpsname[1]{unknown character '#1'}%
145   }%

Generate macros of the form \slotnumber@glyph@{glyphname} that return the
slot number for each glyph. This is faster than parsing \getpsname for the
searched glyph (on the cost of memory).
146   \setcounter{@glyphslot}{0}%
147   \whiledo{\the\c@@glyphslot<256}{%
148     \expandafter\edef
149     \csname slotnumber@glyph@\getpsname{\the\c@@glyphslot}\endcsname{%
150       \the\c@@glyphslot}%
151     \stepcounter{@glyphslot}%
152   }%

Initialise some font-specific things. This is done in a group to save the normal
legend font outside the kerning table.
153   \begingroup

Switch to the font that shall be tested to see if the desired font size is possible etc.
154     \usefont{\krntst@encoding}{\krntst@family}{\krntst@series}{\krntst@shape}%
155     \fontsize{\krntst@size}{\krntst@baselineskip}\selectfont%

Set the Postscript font unit to 0.001 of the design size which is 1 em, normally.
156     \psunit=\krntst@designsize\relax
157     \global\psunit=0.001\psunit

```

Give some feedback.

```
158 \typeout{Requested: \krntst@encoding-\krntst@family-%
159 \krntst@series-\krntst@shape, size \the\krntst@size}%
160 \typeout{Using:\space\space\space\space\space \f@encoding-\f@family-%
161 \f@series-\f@shape, size \f@size pt}%
162 \expandafter\ifdim\the\krntst@size=\f@size pt\relax
163 \else
164 \ClassWarningNoLine{kerntest}{Using different font size than
165 requested}%
166 \fi
167 \setlength{\@tempdima}{\krntst@designsize}%
168 \typeout{Postscript font unit for design size \the\@tempdima:
169 \the\psunit}%
170 \expandafter\ifdim\the\@tempdima=\f@size pt\relax
171 \else
172 \ClassWarningNoLine{kerntest}{The design size (\the\@tempdima,
173 1em by default,\MessageBreak
174 or given value from option 'designsize') of the
175 font\MessageBreak
176 is not equal to the LaTeX font size (\f@size pt).\MessageBreak
177 This can have two reasons:\MessageBreak
178 1. The font does not define 1em to be the design
179 size\MessageBreak
180 \space\space\space (for example, Computer
181 Modern).\MessageBreak
182 2. The font is implicitly scaled by the fd-file\MessageBreak
183 \space\space\space (for example, when using
184 helvet.sty).\MessageBreak
185 This can cause the PostScript font unit length to
186 be\MessageBreak
187 incorrect.
188 You may set the design size for calculation\MessageBreak
189 of the font unit explicitly by using the class\MessageBreak
190 option 'designsize'}%
191 \fi
```

Define the name for the headings and the mtx file (lowercase trick again taken from fontenc.sty).

```
192 \edef\mtxfilename{%
193 \lowercase{\gdef\noexpand\mtxfilename{%
194 \f@encoding-\f@family-\f@series-\f@shape\krntst@extraname}}}%
195 \mtxfilename
```

Set the page headings.

```
196 \markboth{\upshape Font \mtxfilename}{\upshape Font \mtxfilename}%
```

Don't change the page headings by \section etc.

```
197 % \global\def\markboth#1#2{}%
198 % \global\def\markright#1{}%
```

Open the mtx file.

```
199 \typeout{^^JWriting mtx file '\mtxfilename.mtx'^^J}%
200 \immediate\openout\mtxfile\mtxfilename.mtx
```

Write a nice header to the mtx file.

```
201 \protected@write\mtxfile{}{\@percentchar\@percentchar}%
```

```

202 \protected@write\mtxfile{}{\@percentchar\@percentchar\space
203   This is file ‘\mtxfilename.mtx’,}%
204 \protected@write\mtxfile{}{\@percentchar\@percentchar\space
205   generated on \number\year/\number\month/\number\day\space
206   by kernstest.cls, (c) 2004 Harald Harders.}%
207 \protected@write\mtxfile{}{\@percentchar\@percentchar}%
208 \protected@write\mtxfile{}{\@percentchar\@percentchar\space
209   The original source file was:}%
210 \protected@write\mtxfile{}{\@percentchar\@percentchar}%
211 \protected@write\mtxfile{}{\@percentchar\@percentchar\space
212   \jobname\space (.tex?) with these font options:}%
213 \protected@write\mtxfile{}{\@percentchar\@percentchar\space
214   Encoding: \f@encoding}%
215 \protected@write\mtxfile{}{\@percentchar\@percentchar\space
216   Family: \space\space\f@family}%
217 \protected@write\mtxfile{}{\@percentchar\@percentchar\space
218   Series: \space\space\f@series}%
219 \protected@write\mtxfile{}{\@percentchar\@percentchar\space
220   Shape: \space\space\space\f@shape}%
221 \protected@write\mtxfile{}{\@percentchar\@percentchar\space
222   User-defined name: \krntst@extraname}%
223 \protected@write\mtxfile{}{\@percentchar\@percentchar}%
224 \protected@write\mtxfile{}{\string\relax}%
225 \protected@write\mtxfile{}{\string\metrics}%
226 \protected@write\mtxfile{}{\string\needsfontinstversion{1.926}}%
227 \protected@write\mtxfile{}{\@percentchar\@percentchar}%
228 \protected@write\mtxfile{}{\@percentchar\@percentchar\space
229   Kerning data for single characters and}%
230 \protected@write\mtxfile{}{\@percentchar\@percentchar\space
231   the first members of the glyph classes.}%
232 \protected@write\mtxfile{}{\@percentchar\@percentchar}%
233 \protected@write\mtxfile{}{\@percentchar\@percentchar\space
234   After each \string\setkern\space entry, the glyph classes}%
235 \protected@write\mtxfile{}{\@percentchar\@percentchar\space
236   for both glyphs are given (./ means no class).}%
237 \protected@write\mtxfile{}{\@percentchar\@percentchar}%
238 \endgroup
239 }

```

Declare the output handle for the mtx file.

```
240 \newwrite\mtxfile
```

Round a length to an integer value. I am sure this can be done easier, but it works.

```

241 \def\krntst@round#1.#2#3#4\empty{%
242   \setlength\@tempdimc{#1pt}%
243   \if#2.%
244   \else
245     \ifnum#2>4
246       \ifnum#1#2<0
247         \addtolength\@tempdimc{-1.1pt}%
248       \else
249         \addtolength\@tempdimc{1.1pt}%
250     \fi
251   \fi

```

```

252 \fi
253 \edef\rnd@tempa{\strip@pt\@tempdimc}%
254 \expandafter\kernst@round\rnd@tempa.000\@empty
255 }

```

Calculate the rounded length.

```

256 \def\kernst@round#1.#2#3\@empty{\def\kernlen{#1}}

```

**\round** The user routine for rounding lengths. The rounded length is *not* returned but saved in the macro `\kernlen`.

```

257 \newcommand*\round[1]{%
258   \setlength\@tempdimc{#1}%
259   \edef\rnd@tempa{\strip@pt\@tempdimc}%
260   \expandafter\kernst@round\rnd@tempa.000\@empty
261 }

```

Define the Postscript font length.

```

262 \newlength\psunit

```

**\getpsunit** Saves the rounded length of arbitrary unit in Postscript font units in the dimension `\@tempdima`. It has to be used with `\strip@pt` to get rid of the unit “pt” which is wrong of course.

```

263 \newcommand\getpsunit[1]{%
264   \setlength\@tempdima{1pt*\ratio{#1}{\psunit}}%
265 }

```

**\getkern** Get the kerning between the arguments #1 and #2. This is done by typesetting #1#2 with the natural kerning and with suppressed kerning (`#1\kern 0pt#2`). The difference of the box widths is the kerning. Return an integer value in Postscript font units.

```

266 \newcommand\getkern[2]{%
267   \settowidth\@tempdima{#1#2}%
268   \settowidth\@tempdimb{#1\kern0pt#2}%

```

The next line works better than deviding `\@tempdima-\@tempdimb` by 0.001em because rounding errors are avoided.

```

269   \setlength\@tempdima{1pt*\ratio{(\@tempdima-\@tempdimb)*1000}{1em}}%
270   \round{\@tempdima}%
271 }

```

The internal routine for `\saveslotnumber`. Finds out if a slot number or the Postscript name is given and saves the slot number in the counter `@glyphslot`.

```

272 \def@saveslotnumber#1#2\@empty{%
273   \if#1"\relax
274     \setcounter{@glyphslot}{#1#2}%
275   \else
276     \if#1'\relax
277       \setcounter{@glyphslot}{#1#2}%
278     \else
279       \ifnum9<#1\relax
280         \setcounter{@glyphslot}{#1#2}%
281       \else
282         \begingroup\expandafter\expandafter\expandafter\endgroup

```

```

283     \expandafter\ifx\csname slotnumber@glyph@#1#2\endcsname\relax
284     \setcounter{@glyphslot}{-1}%
285     \else
286     \setcounter{@glyphslot}{\csname slotnumber@glyph@#1#2\endcsname}%
287     \fi
288     \fi
289     \fi
290     \fi
291     \ifnum\the\c@glyphslot>255\relax
292     \setcounter{@glyphslot}{-1}%
293     \fi
294 }

```

`\saveslotnumber` Saves the slot number of a glyph given as second argument (by PostScript name or its slot number in decimals, octal, or hexadecimals) in the counter specified in the first argument.

```

295 \DeclareRobustCommand*\saveslotnumber[2]{%
296   \expandafter\@saveslotnumber#2\@empty
297   \setcounter{#1}{\the\c@glyphslot}%
298 }

```

`\getslotnumber` Returns the slot number of a given glyph (by PostScript name or its slot number in decimals, octal, or hexadecimals) in a decimal number.

```

299 \newcommand\getslotnumber[1]{%
300   \expandafter\@saveslotnumber#1\@empty
301   \ifnum\the\c@glyphslot<0\relax
302     \textbf{???}%
303   \else
304     \ifnum\c@glyphslot<100\relax0\fi
305     \ifnum\c@glyphslot<10\relax0\fi
306     \the\c@glyphslot
307   \fi
308 }

```

`\printglyph` Print the glyph with the given PostScript name or slot number (in decimals, octal, or hexadecimals; as usual in L<sup>A</sup>T<sub>E</sub>X). Unfortunately, no kerning appears on the left side of the printed glyph. For example, `\printglyph{A}V` is kerned, but `A\printglyph{V}` isn't. You can solve this by saving the slot number first and by using it later, for example:

```

%\newcounter{slotV}%
%\saveslotnumber{slotV}{V}%
%A\char\arabic{slotV}
%
309 \newcommand*\printglyph[1]{%
310   \expandafter\@saveslotnumber#1\@empty
311   \char\the\c@glyphslot
312 }

```

A help macro for comparing arguments with “-”.

```

313 \edef\@minussign{-}%

```

Counters storing the slot numbers for the three glyphs used within one line of the `kerntable` environment.

```
314 \newcounter{@slota}
315 \newcounter{@slotb}
316 \newcounter{@slotc}
```

`\testkern` The main macro of the class. It takes 5 arguments:  
{*glyph 1*}{*kerneling 1-2*}{*glyph 2*}{*kerneling 2-3*}{*glyph 3*}.

The glyphs are given by their number, not the glyphs itself.

```
317 \newcommand\testkern[5]{%
```

Save the kerning arguments globally because otherwise they got lost from tabular cell to tabular cell.

```
318 \xdef\@kernlena{#2}%
319 \xdef\@kernlenb{#4}%
```

Get the slot numbers for the three characters and save them in the counters `@slota`, `@slotb`, and `@slotc`.

```
320 \saveslotnumber{@slota}{#1}%
321 \ifnum\the\c@@slota<0%
322 \ClassError{kerntest}{Used unknown glyph '#1'}{%
323 You may have misspelled the glyph or you have taken an invalid
324 number.}%
325 \setcounter{@slota}{0}%
326 \fi
327 \saveslotnumber{@slotb}{#3}%
328 \ifnum\the\c@@slotb<0%
329 \ClassError{kerntest}{Used unknown glyph '#3'}{%
330 You may have misspelled the glyph or you have taken an invalid
331 number.}%
332 \setcounter{@slotb}{0}%
333 \fi
334 \saveslotnumber{@slotc}{#5}%
335 \ifnum\the\c@@slotc<0%
336 \ClassError{kerntest}{Used unknown glyph '#5'}{%
337 You may have misspelled the glyph or you have taken an invalid
338 number.}%
339 \setcounter{@slotc}{0}%
340 \fi
```

Find out if there are old kerning data for one of the two glyph pairs.

First pair.

The better form of `\@ifundefined` that does not define its argument as side-effect.

```
341 \begingroup\expandafter\expandafter\expandafter\endgroup
342 \expandafter\ifx\csname kt@kerning@\getpsname{\the\c@@slota}@%
343 \getpsname{\the\c@@slotb}\endcsname\relax
```

No old kerning. Thus don't do any kerning later.

```
344 \gdef\oldkerna{}%
345 \else
```

Old kerning exists. Save the old kerning to apply it later.

```
346 \gdef\oldkerna{%
347 \kern
```

```

348     \csname kt@kerning@\getpsname{\the\c@@slota}%
349     \getpsname{\the\c@@slotb}\endcsname
350     \psunit
351   }%

  If no new kerning ist given just tell the user that he reuses a kerning.
352   \ifx\@kernlena\@minussign
353     \typeout{Kerning pair for \getpsname{\the\c@@slota}-%
354     \getpsname{\the\c@@slotb} reused (value
355     \csname kt@kerning@\getpsname{\the\c@@slota}%
356     \getpsname{\the\c@@slotb}\endcsname).}%
357   \else

  Old kerning exists and new kerning, too. Test if the old and new kernings are
  identical.
358     \ifnum\@kernlena=\csname kt@kerning@\getpsname{\the\c@@slota}%
359     \getpsname{\the\c@@slotb}\endcsname\relax

  Yes. Nevertheless, generate a warning.
360     \ClassWarning{kerntest}{Kerning for
361     \getpsname{\the\c@@slota}-\getpsname{\the\c@@slotb}\MessageBreak
362     repeated (value #2)}%
363   \else

  No. Produce an erroe message.
364     \ClassError{kerntest}{Conflicting kerning for
365     \getpsname{\the\c@@slota}-\getpsname{\the\c@@slotb}\MessageBreak
366     (new value #2 != old value
367     \csname kt@kerning@\getpsname{\the\c@@slota}%
368     \getpsname{\the\c@@slotb}\endcsname)%
369   }{%
370     You have given the kerning for the specified glyph pair
371     twice with different\MessageBreak
372     values. This may also appear when using glyph classes.
373     You have to give the\MessageBreak
374     kerning only once per glyph class.\MessageBreak
375     You may leave out the second kerning pair, or you may
376     give\MessageBreak
377     the kerning '-'. Then, the old value is reused.
378   }%
379   \fi
380   \fi
381   \fi

  Second pair.
382   \begingroup\expandafter\expandafter\expandafter\endgroup
383   \expandafter\ifx\csname kt@kerning@\getpsname{\the\c@@slotb}%
384   \getpsname{\the\c@@slotc}\endcsname\relax
385     \gdef\oldkernb{%
386   \else
387     \gdef\oldkernb{%
388       \kern
389       \csname kt@kerning@\getpsname{\the\c@@slotb}%
390       \getpsname{\the\c@@slotc}\endcsname
391       \psunit
392     }%

```

```

393 %
394 \ifx\@kernlenb\@minussign
395 \typeout{Kerning pair for \getpsname{\the\c@@slotb}-%
396 \getpsname{\the\c@@slotc} reused (value
397 \csname kt@kerning@\getpsname{\the\c@@slotb}@%
398 \getpsname{\the\c@@slotc}\endcsname).}%
399 \else
400 \ifnum\@kernlenb=\csname kt@kerning@\getpsname{\the\c@@slotb}@%
401 \getpsname{\the\c@@slotc}\endcsname\relax
402 \ClassWarning{kerntest}{Kerning for
403 \getpsname{\the\c@@slotb}-\getpsname{\the\c@@slotc}\MessageBreak
404 repeated (value #4)}%
405 \else
406 \ClassError{kerntest}{Conflicting kerning for
407 \getpsname{\the\c@@slotb}-\getpsname{\the\c@@slotc}\MessageBreak
408 (new value #4 != old value
409 \csname kt@kerning@\getpsname{\the\c@@slotb}@%
410 \getpsname{\the\c@@slotc}\endcsname)%
411 }{%
412 You have given the kerning for the specified glyph pair
413 twice with different\MessageBreak
414 values. This may also appear when using glyph classes.
415 You have to give the\MessageBreak
416 kerning only once per glyph class.\MessageBreak
417 You may leave out the second kerning pair, or you may
418 give\MessageBreak
419 the kerning '-'. Then, the old value is reused.
420 }%
421 \fi
422 \fi
423 \fi

First, type the slot number of glyph 2.
424 \krntst@helpfont\getslotnumber{#3}%
425 &

Type the postscript name of glyph 2.
426 \krntst@helpfont\getpsname{\the\c@@slotb}%
427 &

Print the three glyphs with original kerning.
428 \char\the\c@@slota\char\c@@slotb\char\c@@slotc%
429 &

Print glyph 1.
430 \char\the\c@@slota%

If a kerning is given, apply it; otherwise do nothing.
431 \ifx\@kernlena\@minussign
432 \oldkerna
433 \else
434 \kern#2\psunit
435 \fi

Print glyph 2.
436 \char\the\c@@slotb%

```

If a kerning is given, apply it; otherwise do nothing.

```
437 \ifx\@kernlenb\@minussign
438   \oldkernb
439 \else
440   \kern#4\psunit
441 \fi
```

Print glyph 3.

```
442 \char\the\c@@slotc%
443 &
```

Do the same as in columns 2 and 3, but twice at the same place. First, natural kerning.

```
444 \color{oldcolor}%
445 \makebox[0pt][l]{\char\the\c@@slota\char\the\c@@slotb\char\the\c@@slotc}%
```

Second, newly given kerning. Switch the color depending if a kerning has been given.

```
446 \ifx\@kernlena\@minussign
447   \ifthenelse{equal{\oldkerna}{}}{\color{newcolor}}%
448 \else
449   \color{newcolor}%
450 \fi
451 \ifx\@kernlenb\@minussign
452   \ifthenelse{equal{\oldkernb}{}}{\color{newcolor}}%
453 \else
454   \color{newcolor}%
455 \fi
456 \char\the\c@@slota%
457 \ifx\@kernlena\@minussign
458   \oldkerna
459 \else
460   \kern#2\psunit
461 \fi
462 \char\the\c@@slotb%
463 \ifx\@kernlenb\@minussign
464   \oldkernb
465 \else
466   \kern#4\psunit
467 \fi
468 \char\the\c@@slotc%
469 &
```

Get the value of the natural kerning. This has to be done with the tested font switched on to get the right values. This value is saved in `\kernlen` for later use.

```
470 \getkern{\char\the\c@@slota}{\char\the\c@@slotb}%
```

Switch to the legend font.

```
471 \krntst@helpfont
```

If no kerning is given (`{\kerning 1-2}`==) print out the original kerning. The part `\ifdim... \fi` adds a - if the kerning is negative. Together with the negative kerning, this gives a “-” instead of a “.”.

```
472 \ifx\@kernlena\@minussign
473   \ifthenelse{equal{\oldkerna}{}}{%
474     \textcolor{oldcolor}{\small
```

```

475     \ifdim\kernlen pt<Opt-\fi
476     \ifdim\kernlen pt>Opt+\fi
477     \kernlen}%
478 }{%
479   \ifnum\kernlen=0\relax
480   \else
481     \textcolor{oldcolor}{\small(
482     \ifdim\kernlen pt<Opt-\fi
483     \ifdim\kernlen pt>Opt+\fi
484     \kernlen)}}%
485   \fi
486   ~\textcolor{newcolor}{\large
487   \ifnum
488     \csname kt@kerning@\getpsname{\the\c@@slota}%
489     \getpsname{\the\c@@slotb}\endcsname<0-\fi
490     \csname kt@kerning@\getpsname{\the\c@@slota}%
491     \getpsname{\the\c@@slotb}\endcsname
492     \makebox[Opt][l]{\$^\ast$}%
493   }%
494 }%

```

Write old kerning to mtx file.

```

495   \ifthenelse{\boolean{krntst@writeall}\and\not\equal{\kernlen}{0}}{%
496     \writemtxkern[original kerning]{\the\c@@slota}{%
497     \ifdim\kernlen pt>Opt+\fi\kernlen}{\the\c@@slotb}%
498   }{}%

```

If a kerning is given print the new kerning (same trick with negative numbers).

```

499   \else

```

If there were original kerning data, print the in parenthesis first.

```

500   \ifdim\kernlen pt=Opt
501   \else
502     \textcolor{oldcolor}{\small(
503     \ifdim\kernlen pt<Opt-\fi
504     \ifdim\kernlen pt>Opt+\fi
505     \kernlen)}
506   \fi
507   \textcolor{newcolor}{\large
508   \ifnum#2<0-\fi#2%
509   \ifthenelse{\equal{\oldkerna}{}}{%
510     }\makebox[0mm][l]{\$^\dagger$}%
511   }%

```

Write the new kerning information into the mtx file.

```

512   \ifthenelse{\equal{\oldkerna}{}}{%
513     \writemtxkern{\the\c@@slota}{#2}{\the\c@@slotb}%
514   }{}%
515   \fi
516   &

```

Do the same for the second kerning pair.

```

517   \getkern{\char\the\c@@slotb}{\char\the\c@@slotc}%
518   \krntst@helpfont
519   \ifx\@kernlenb\@minussign
520   \ifthenelse{\equal{\oldkernb}{}}{%

```

```

521     \textcolor{oldcolor}{\small
522     \ifdim\kernlen pt<Opt-\fi
523     \ifdim\kernlen pt>Opt+\fi
524     \kernlen}%
525 }{%
526   \ifnum\kernlen=0\relax
527   \else
528     \textcolor{oldcolor}{\small(
529     \ifdim\kernlen pt<Opt-\fi
530     \ifdim\kernlen pt>Opt+\fi
531     \kernlen)}%
532   \fi
533   ~\textcolor{newcolor}{\large
534   \ifnum
535     \csname kt@kerning@\getpsname{\the\c@slotb}%
536     \getpsname{\the\c@slotc}\endcsname<0-\fi
537     \csname kt@kerning@\getpsname{\the\c@slotb}%
538     \getpsname{\the\c@slotc}\endcsname
539     \makebox[Opt][l]{\$^\ast$}%
540   }%
541 }%
542 \ifthenelse{\boolean{krntst@writeall}\and\not\equal{\kernlen}{0}}{%
543   \writetxkern[original kerning]{\the\c@slotb}{%
544     \ifdim\kernlen pt>Opt+\fi\kernlen}{\the\c@slotc}%
545 }{%
546   \else
547     \ifdim\kernlen pt=0pt
548     \else
549       \textcolor{oldcolor}{\small(
550       \ifdim\kernlen pt<Opt-\fi
551       \ifdim\kernlen pt>Opt+\fi
552       \kernlen)}
553     \fi
554     \textcolor{newcolor}{\large
555     \ifnum#4<0-\fi#4%
556     \ifthenelse{\equal{\oldkernb}{}}{%
557       }{\makebox[0mm][l]{\$^\dagger$}%
558     }%
559     \ifthenelse{\equal{\oldkernb}{}}{%
560       \writetxkern{\the\c@slotb}{#4}{\the\c@slotc}%
561     }{%
562     \fi
563     &

```

Print the example with natural kerning.

```

564   \char\the\c@slota\char\the\c@slotb
565   \krntst@example
566   \char\the\c@slotb\char\the\c@slotc
567   &

```

Print the example with new kerning.

```

568   \char\the\c@slota%
569   \ifx\@kernlena\@minussign
570   \else
571     \kern#2\psunit

```

```

572 \fi
573 \char\the\c@@slotb\krntst@example\char\the\c@@slotb
574 \ifx\@kernlenb\@minussign
575 \else
576 \kern#4\psunit
577 \fi
578 \char\the\c@@slotc%
579 &

```

Switch to legend font for the comments that may appear.

```

580 \krntst@helpfont\ignorespaces
581 }

```

**kerntable** The kerning table environment.

```

582 \newenvironment{kerntable}{%

```

Switch to the tested font.

```

583 \usefont{\krntst@encoding}{\krntst@family}{\krntst@series}{\krntst@shape}%
584 \fontsize{\krntst@size}{\krntst@baselineskip}\selectfont%

```

Start a `longtable` environment for the kerning samples.

```

585 \begin{longtable}[l]{@{}l1l@{~}l@{~}l@{~}rrl1l@{}}

```

Type the header of the table.

```

586 \krntst@helpfont slot&
587 \krntst@helpfont name&
588 \krntst@helpfont orig&
589 \krntst@helpfont new&
590 \krntst@helpfont both&
591 \krntst@helpfont k.\,1&
592 \krntst@helpfont k.\,2&
593 \krntst@helpfont orig.&
594 \krntst@helpfont new&
595 \krntst@helpfont comment\\
596 \endhead

```

Repeat it as footer.

```

597 \krntst@helpfont slot&
598 \krntst@helpfont name&
599 \krntst@helpfont orig&
600 \krntst@helpfont new&
601 \krntst@helpfont both&
602 \krntst@helpfont k.\,1&
603 \krntst@helpfont k.\,2&
604 \krntst@helpfont orig.&
605 \krntst@helpfont new&
606 \krntst@helpfont comment\\
607 \endfoot
608 }{%

```

And now the end of the table.

```

609 \end{longtable}%
610 \ignorespacesafterend
611 }

```

```

\writemtxkern Write an entry into the mtx file. This command copies double quotes to single
quotes if requested (only if no optional argument is given).
612 \newif\if@tempswb
613 \newcommand\writemtxkern[4][\@empty]{%
Store the glyph names of both glyphs in \@firstglyphname and \@secondglyphname.
614 \expandafter\@saveslotnumber#2\@empty
615 \edef\@firstglyphname{\getpsname{\c@glyphslot}}%
616 \expandafter\@saveslotnumber#4\@empty
617 \edef\@secondglyphname{\getpsname{\c@glyphslot}}%
Test if a comment has been given.
618 \ifthenelse{equal{#1}{\@empty}}{%
Get the corresponding glyph class for the first character and save it in \rightkern.
If none, \rightkern is set to \@empty.
619 \edef\rightkern{\getclassofglyph{right}{\@firstglyphname}}%
If the glyph is in no glyph class, make a temporary glyph class \rightkern which
contains only this glyph. Define the comment \textright for the mtx file.
620 \ifthenelse{equal{\rightkern}{\@empty}}{%
621 \edef\textright{./}%
622 \def\rightkern{\@firstglyphname}%
623 }{%
624 \edef\textright{\expandafter\@getclassname\rightkern\@empty}%
625 }%
Get the corresponding glyph class for the second character and save it in
\leftkern. If none, \leftkern is set to \@empty.
626 \edef\leftkern{\getclassofglyph{left}{\@secondglyphname}}%
If the glyph is in no glyph class, make a temporary glyph class \leftkern which
contains only this glyph. Define the comment \leftkern for the mtx file.
627 \ifthenelse{equal{\leftkern}{\@empty}}{%
628 \edef\leftkern{./}%
629 \def\leftkern{\@secondglyphname}%
630 }{%
631 \edef\leftkern{\expandafter\@getclassname\leftkern\@empty}%
632 }%
Set the kernig data for all kerning pairs that can be found in both glyph classes
\rightkern and \leftkern.
633 \@tempwbtrue
634 \@forallinclass{\rightkern}{first}{%
635 \@forallinclass{\leftkern}{second}{%
Write the kerning data to the mtx file, but only for the first members of the
glyph classes. The others are set in the mtx file by \setrightkerning and
\setleftkerning.
636 \if@tempwb
637 \protected@write\mtxfile{}{%
638 \string\setkern
639 {\first}{\second}{#3}%
640 \@percentchar\space\space\textright\space-- \textleft
641 }% \protected@write
642 \@tempwbfalse
643 \fi

```

Define a command `\kt@kerning@⟨first glyph⟩@⟨second glyph⟩` that contains the kerning for later testing on conflicting values. Scale the kerning data according to the given values in `\defglyphclass`.

```

644     \setcounter{@tmpscale}{#3*\first@scaling*\second@scaling/1000000}%
645     \typeout{\first-\second: \the@tmpscale}%
646     \expandafter\xdef\csname kt@kerning@⟨first @⟨second⟩\endcsname{%
647         \the@tmpscale}%
648 %     \expandafter\xdef\csname kt@kerning@⟨first @⟨second⟩\endcsname{#3}%
649     }% forallinclass second
650 }% forallinclass first
651 }-%

```

If an optional argument has been given, just write this kerning pair without any tests.

```

652     \protected@write\mtxfile{}{-%
653         \string\setkern
654         {⟨firstglyphname⟩{⟨secondglyphname⟩{#3}%
655         \@percentchar\space\space #1%
656     }%

```

Nevertheless, generate the command for testing on conflicting values.

```

657     \expandafter\xdef
658     \csname kt@kerning@⟨firstglyphname⟩@⟨secondglyphname⟩\endcsname{#3}%
659 }%
660 }%

```

### 6.1.1 Glyph classes

`\defglyphclass` The macro `\defglyphclass{⟨side⟩}{⟨name⟩}{⟨glyphlist⟩}` defines a class of glyphs that have the same kerning on the same `⟨side⟩` which has to be “left” or “right”. `⟨name⟩` is the name of the glyph class while `⟨glyphlist⟩` is a comma-separated list of all glyphs that have the same kerning on their `⟨side⟩` side.

```

661 \newcounter{@tmpscale}
662 \newcounter{@firstscale}
663 \newcommand\defglyphclass[3]{%

```

Do it at `\begin{document}` because otherwise it is not clear which encoding is used and thus the glyphs are not yet known.

```

664 \AtBeginDocument{%

```

Test if a list of glyph classes exists for the chosen `⟨side⟩`.

```

665     \@ifundefined{glyphclasslist@#1}{-%

```

No glyph class of the current `⟨side⟩` has been defined, yet. Install a new one.

```

666     \expandafter\def\csname glyphclasslist@#1\endcsname{%
667         glyphclass@#1@#2}%
668     }-%

```

The needed glyph-class list exists. Test if there is an old glyph class with the same name (`⟨side⟩` and `⟨name⟩`).

```

669     \begingroup
670     \@tempwatrue
671     \forallclasses{#1}{@tmpcls}{%
672         \ifthenelse{\equal{\@tmpcls}{glyphclass@#1@#2}}{-%
673         \@tempwafalse

```

```

674     }{}%
675   }%

```

If this is not the case, append the new glyph class to the glyph class list.

```

676   \if@tempswa
677     \expandafter\xdef\csname glyphclasslist@#1\endcsname{%
678     \csname glyphclasslist@#1\endcsname,glyphclass@#1@#2}%
679   \fi
680   \endgroup
681 }%

```

Define the macro `\glyphclass@<side>@<name>` that stores the *<glyphlist>* for this glyph class. At this stage it is defined empty in order to avoid that error messages are generated for “already used glyphs”.

```

682   \expandafter\def\csname glyphclass@#1@#2\endcsname{}%

```

Store the new *<glyphlist>* in a temporary variable `\tmpglyphclass`. To do this, all glyphs of the list are converted to Postscript glyph names and tested if they are valid. Also, it is tested if a glyph is contained double.

```

683   \edef\@tempa{#3}%
684   \@tempwbtrue
685   \@forallinclass{@tempa}{@tmpglyph}{%
686     \saveslotnumber{@glyphslot}{\@tmpglyph}%
687     \ifnum\the\c@@glyphslot<0%
688       \ClassError{kerntest}{Used unknown glyph ‘\@tmpglyph’}{%
689         You may have misspelled the glyph or you have taken an invalid
690         number.
691         This incorrect glyph is ignored.}%
692     \edef\thisglyphname{???}%
693   \else
694     \edef\thisglyphname{\getpsname{\the\c@@glyphslot}}%
695     \if@tempswb
696       \global\@tempswbfalse
697       \setcounter{@firstscale}{\@tmpglyph@scaling}%
698     \fi
699     \setcounter{@tmpscale}{1000*\@tmpglyph@scaling/\the@firstscale}%

```

Now, it has to be tested if none of the glyphs of the new glyph list are in this or another list already. If so, generate an error message. Save the error state in `\@tempswa` to be able to add the glyph only if it is in no other glyph class.

```

700   \@tempwatrue
701   \forallclasses{#1}{@tmpcls}{%
702     \@ifglyphinclass{\@tmpcls}{\@tmpglyph}{%
703       \@tempswafalse
704       \ClassError{kerntest}{Glyph ‘\@tmpglyph’
705         (‘\thisglyphname’,\MessageBreak
706         glyph class #1/#2) already\MessageBreak
707         in glyph
708         class (\expandafter\getclassname\@tmpcls\@empty)}%
709     }{%
710       Each glyph may only be once in one glyph class for
711       each side.
712     }%
713   }{}%
714 }%

```

Append this glyph to the current glyph list.

```

715     \if@tempswa
716     \ifthenelse{\equal{\csname glyphclass@#1@#2\endcsname}{}}{%
717         \expandafter\edef\csname glyphclass@#1@#2\endcsname{%
718             \thisglyphname[\the@tmpscale]}%
719         \edef\firstglyphinclass{\thisglyphname}%
720     }{%
721         \expandafter\edef\csname glyphclass@#1@#2\endcsname{%
722             \csname glyphclass@#1@#2\endcsname,%
723             \thisglyphname[\the@tmpscale]}%
724     }%

```

Generate a macro `\glyphclass@glyph@glyphname` which saves the corresponding glyph class for each glyph for faster access.

```

725     \expandafter\edef
726     \csname glyphclass@glyph@#1@\thisglyphname\endcsname{%
727         glyphclass@#1@#2}%
728     \fi
729     \fi
730 }%

```

Some feedback.

```

731     \typeout{Glyph class '#1/#2' (\csname glyphclass@#1@#2\endcsname)
732         defined.}%
733 }%
734 }

```

`\newglyphclass` The macro `\newglyphclass` works as `\defglyphclass` but defines a new glyph class. It produces an error if the class already exists.

```

735 \newcommand\newglyphclass[3]{%

```

Test if this glyph class already exists and generate an error message if so. If not, call `\defglyphclass` to save the new glyph class.

```

736 \AtBeginDocument{%
737     \@tempswatrue
738     \forallclasses{#1}{@tmpcls}{%
739         \ifthenelse{\equal{\@tmpcls}{glyphclass@#1@#2}}{%
740             \ClassError{kerntest}{Class '#1/#2' already exists}{%
741                 The command is ignored.}%
742         \@tempswafalse
743     }{%
744     }%
745     \if@tempswa
746         \defglyphclass{#1}{#2}{#3}%
747     \fi
748 }%
749 }

```

`\renewglyphclass` The macro `\renewglyphclass` works as `\newglyphclass` but redefines an existing one.

```

750 \newcommand\renewglyphclass[3]{%

```

Test if this glyph class does not exist and generate an error message if so. If it exists, call `\defglyphclass` to redefine the glyph class.

```

751 \AtBeginDocument{%

```

```

752 \@tempswafalse
753 \forallclasses{#1}{@tmpcls}{%
754 \ifthenelse{\equal{\@tmpcls}{glyphclass@#1@#2}}{%
755 \@tempswatrue
756 }{}}%
757 }%
758 \if@tempswa
759 \defglyphclass{#1}{#2}{#3}%
760 \else
761 \ClassError{kerntest}{Class ‘#1/#2’ does not exists}{%
762 The command is ignored.}%
763 \fi
764 }%
765 }

```

`\provideglyphclass` The macro `\provideglyphclass` works as `\newglyphclass` but does only do its job if the glyph class does not exist right now.

```

766 \newcommand\provideglyphclass[3]{%
    Test if this glyph class already exists. If not, call \defglyphclass to save the new
    glyph class.

```

```

767 \AtBeginDocument{%
768 \@tempswatrue
769 \forallclasses{#1}{@tmpcls}{%
770 \ifthenelse{\equal{\@tmpcls}{glyphclass@#1@#2}}{%
771 \@tempswafalse
772 }{}}%
773 }%
774 \if@tempswa
775 \defglyphclass{#1}{#2}{#3}%
776 \fi
777 }%
778 }

```

Type out the human readable name *(side)/(name)* for a glyph class, giving the name of the corresponding macro. No test on a correct name is made.

```

779 \def\@getclassname#1@#2@#3\@empty{#2/#3}

```

`\getclassofglyph` Syntax: `\getclassofglyph{<side>}{<glyph name>}`.

Return the name of the glyph class, that contains the argument. If it is not contained in any class, `\@empty` is returned. The glyph name has to be given as argument.

```

780 \newcommand*\getclassofglyph[2]{%
781 \expandafter\ifx\csname glyphclass@glyph@#1@#2\endcsname\relax
782 \@empty
783 \else
784 \csname glyphclass@glyph@#1@#2\endcsname
785 \fi
786 }

```

An internal boolean for searching glyph classes.

```

787 \newif\if@glyphfound

```

`\ifglyphinclass` Syntax: `\ifglyphinclass{<side>}{<name>}{<glyph>}{<yes>}{<no>}`. Tests if the glyph `<glyph>` is contained in the glyph class `<side>/<name>`. Depending on that, `<yes>` or `<no>` are executed. The work is done by `\@ifglyphinclass` described later.

```

788 \newcommand\ifglyphinclass[5]{%
789   \@ifundefined{glyphclass@#1@#2}{%
790     \ClassError{kerntest}{Glyph class #1/#2 not available}{}%
791   }{%
792     \edef\@tempa{#3}%
793     \saveslotnumber{@tempcnta}{\@tempa}%
794     \ifnum\the\c@tempcnta<0%
795       \ClassError{kerntest}{Used unknown glyph ‘#2’}{%
796         You may have misspelled the glyph or you have taken an invalid
797         number.}%
798     \else
799       \ifthenelse{\equal{glyphclass@#1@#2}{%
800         \getclassofglyph{#1}{\getpsname{\the\c@tempcnta}}}}{%
801         #4%
802       }{%
803         #5%
804       }%
805     \fi
806 %   \@ifglyphinclass{glyphclass@#1@#2}{#3}{#4}{#5}%
807 }%
808 }

```

`\@ifglyphinclass` Syntax: `\@ifglyphinclass{<macro>}{<glyph>}{<yes>}{<no>}`. Tests if the glyph `<glyph>` is contained in the glyph class with the macro name `\<macro>`. Depending on that, `<yes>` or `<no>` are executed.

```

809 \newcounter{@tempcnta}
810 \newcommand\@ifglyphinclass[4]{%
811   \@ifundefined{#1}{%
812     \ClassError{kerntest}{Glyph class #1 not available}{}%
813   }{%

```

Extract the name of the side from the class macro name.

```

814   \def\krntst@side##1@##2@##3\@empty{\edef\krntst@side{##2}}%
815   \expandafter\krntst@side#1\@empty

```

Extract the name of the side from the class macro name.

```

816   \edef\@tempa{#2}%

```

Find out if the glyph is valid.

```

817   \saveslotnumber{@tempcnta}{\@tempa}%
818   \ifnum\the\c@tempcnta<0\relax
819     \ClassError{kerntest}{Used unknown glyph ‘#2’}{%
820       You may have misspelled the glyph or you have taken an invalid
821       number.}%
822   \else

```

Test if the corresponding class to the glyph is the requested one.

```

823     \ifthenelse{\equal{#1}{%
824       \getclassofglyph{\krntst@side}{\getpsname{\the\c@tempcnta}}}}{%
825       #3%
826     }{%

```

```

827         #4%
828     }%
829     \fi
830 }%
831 }

```

`\forallinclasses` Syntax: `\forallinclass{<side>}{<name>}{<glyph>}{<action>}`

The commands in `<action>` are executed once for every glyph of the glyph class `<side>/<name>`. In each run, the specific glyph is stored in the macro `\<glyph>` which has to be given without leading backslash. This routine can be nested if `<glyph>` is different for both layers, e.g.,

```

%\forallinclass{left}{H}{outer}{%
% \forallinclass{left}{H}{inner}{%
%   glyph pair: ‘‘\outer’’--‘‘\inner’’\
% }%
%}
%

832 \newcommand\forallinclass[4]{%
833   \@forallinclass{glyphclass@#1@#2}{#3}{#4}%
834 }

```

`@forallinclasses` Syntax: `\@forallinclass{<macro>}{<glyph>}{<action>}`

The internal command for `\forallinclasses`. Takes the macro name for the glyph class instead of the side and the name.

```

835 \newcommand\@forallinclass[3]{%
    Redefine \stoploop to use the current glyph variable as default.
836   \renewcommand\stoploop[1][#2]{%
837     \expandafter\edef\csname #1\endcsname{%
838       }%
    Get the first glyph of the glyph list. It is stored in \<glyph>.
839   \begingroup\expandafter\expandafter\expandafter\endgroup
840   \expandafter\ifx\csname #1\endcsname\relax
841     \edef\kt@tempa{%
842       \else
843       \edef\kt@tempa{\csname #1\endcsname}%
844       \fi
845   \expandafter\@nextglyphinclass\kt@tempa,\@empty{#2}%
    If this glyph is not empty, the end of the glyph class has not been reached. Then,
    enter the loop.
846   \whiledo{\not\equal{\csname #2\endcsname}{}}{%
    Execute the loop commands.
847     #3%
    Get the next glyph of the glyph list. It is stored in \<glyph>. The if clause is
    necessary to handle \stoploop.
848     \ifthenelse{\equal{\csname #2\endcsname}{}}{%
849       }{%
850       \edef\kt@tempa{\csname #2@rest\endcsname}%
851       \expandafter\@nextglyphinclass\kt@tempa,\@empty{#2}%
852     }%

```

```
853 }%
854 }
```

`\@nextglyphinclass` Everything before the first comma in the list is the next glyph in the glyph class. Store it in the macro given at the end of the argument list ( $\langle\#3\rangle$ ). Store the rest of the glyph class in  $\langle\#3\rangle$ @rest for later work on it.

```
855 \def\@parseglyphname#1[#2]#3\@empty#4{%
856   \expandafter\edef\csname#4\endcsname{#1}%
857   \expandafter\xdef\csname#4@scaling\endcsname{#2}%
858 %   \typeout{--> '#1' mit [\csname #4@scaling\endcsname], ignoriert: '#3', Name: '#4'}%
859 }
860 \def\@nextglyphinclass#1,#2\@empty#3{%
861   \expandafter\edef\csname#3@rest\endcsname{#2}%
862   \expandafter\@parseglyphname#1[1000]\@empty{#3}%
863 }
```

`\forallclasses` Syntax: `\forallclasses{<side>}{<glyph class>}{<action>}`

The commands in  $\langle action \rangle$  are executed once for every glyph class of the glyph-class list  $\langle side \rangle$ . In each run, the specific glyph class is stored in the macro  $\langle glyph class \rangle$  which has to be given without leading backslash. This routine can be nested if  $\langle glyph class \rangle$  is different for both layers. (Same as `\forallinclass`).

```
864 \newcommand\forallclasses[3]{%
865   \renewcommand\stoploop[1][#2]{%
866     \expandafter\edef\csname ##1\endcsname{%
867       }%
868     \begingroup\expandafter\expandafter\expandafter\endgroup
869     \expandafter\ifx\csname glyphclasslist@#1\endcsname\relax
870     \edef\kt@tempa{%
871       \else
872       \edef\kt@tempa{\csname glyphclasslist@#1\endcsname}%
873     \fi
874     \expandafter\@nextclass\kt@tempa,\@empty{#2}%
875     \whiledo{\not\equal{\csname #2\endcsname}{}}{%
876       #3%
877       \ifthenelse{\equal{\csname #2\endcsname}{}}{%
878         }{%
879           \edef\kt@tempa{\csname #2@rest\endcsname}%
880           \expandafter\@nextclass\kt@tempa,\@empty{#2}%
881         }%
882       }%
883 }
```

`\stoploop` Stops the execution of `\forallclasses` or `\forallinclass`. The optional argument gives the stop variable. By this, also the outer loop can be stopped from the inner one.

```
884 \newcommand\stoploop[1][ ]{%
885   \expandafter\edef\csname #1\endcsname{%
886 }
```

`\@nextglyphinclass` Everything before the first comma in the list is the next glyph class in the glyph-class list. Store it in the macro given at the end of the argument list ( $\langle\#3\rangle$ ). Store the rest of the glyph-class list in  $\langle\#3\rangle$ @rest for later work on it.

```
887 \def\@nextclass#1,#2\@empty#3{%
```

```

888 \expandafter\edef\csname #3@rest\endcsname{#2}%
889 \expandafter\edef\csname #3\endcsname{#1}%
890 }

```

### 6.1.2 Extra commands for special encodings

Setup command for the different encodings.

```

891 \newcommand\encodingsetup[1]{%
892 \AtBeginDocument{%
893 \typeout{Setup for font encoding.
894 This differs from encoding to encoding.}%
895 \setkeys{krnenc}{#1}%
896 }%
897 }
898 \@onlypreamble\encodingsetup

```

## 6.2 Footer of mtx file

Write a footer to the `mtx` file. This is done as last action of the class in order to ensure that all other things have been done before.

```
899 \AtEndDocument{%
```

First, write the kerning data for the glyph classes.

```

900 \typeout{Writing kerning factors for glyph classes to mtx file}%
901 \protected@write\mtxfile{}{\@percentchar\@percentchar}%
902 \protected@write\mtxfile{}{\@percentchar\@percentchar\space
903 Kerning factors for the different glyph classes.}%
904 \protected@write\mtxfile{}{\@percentchar\@percentchar}%

```

Define a command that does the output for the different sides.

```

905 \def\@writeclass#1{%
906 \forallclasses{#1}{@tmpcls}{%
907 \typeout{\space\space for class
908 \expandafter\@getclassname\@tmpcls\@empty}%

```

The first glyph is the referent glyph of the class. If the inner loop is executed the first time, set `\firstglyphinclass` to this value.

```

909 \@tempswatrue
910 \@forallinclass{\@tmpcls}{@tmpglyph}{%
911 \if@tempswa
912 \edef\firstglyphinclass{\@tmpglyph}%
913 \@tempswafalse
914 \else

```

For the other members of this class, write the `\set<side>kerning` commands to the `mtx` file.

```

915 \protected@write\mtxfile{}{%
916 \string\set#1kerning%
917 {\@tmpglyph}{\firstglyphinclass}{\@tmpglyph@scaling}%
918 \@percentchar\space\space
919 \expandafter\@getclassname\@tmpcls\@empty
920 }%
921 \fi
922 }%
923 }%

```

```

924 }%
    The sides left and right are defined.
925 \@writeclass{left}%
926 \protected@write\mtxfile-{\@percentchar\@percentchar}%
927 \@writeclass{right}%
    Write a real footer.
928 \protected@write\mtxfile-{\@percentchar\@percentchar}%
929 \protected@write\mtxfile-{\string\endmetrics}%
930 \protected@write\mtxfile-{\@percentchar\@percentchar}%
931 \protected@write\mtxfile-{\@percentchar\@percentchar\space
932   End of file ‘\mtxfilename.mtx’.%
933 \closeout\mtxfile
934 \typeout{^^JWritten mtx file ‘\mtxfilename.mtx’^^J}%
935 }
936 </class>

```

### 6.3 Class option files

To be able to write correct `mtx` files, the class has to know which glyph number has which Postscript name. This is done by the `\getpsname` macro which depends on the used encoding. This is done by loading different class option files.

#### 6.3.1 T1 encoding

The T1 encoding. The data are taken from `t1.etx`.

```

937 (*mtx & t1)
938 \makeatletter
    Set options to switch to other font shapes.
939 \define@key{krnenc}{ligaturing}[1]{%
940   \typeout{T1 encoding: ligaturing = #1}%
941   \ifnum#1<0\relax
942     \expandafter\def\csname krntst@T1@027\endcsname{.notdef.027}%
943     \expandafter\def\csname krntst@T1@028\endcsname{.notdef.028}%
944     \expandafter\def\csname krntst@T1@029\endcsname{.notdef.029}%
945     \expandafter\def\csname krntst@T1@030\endcsname{.notdef.030}%
946     \expandafter\def\csname krntst@T1@031\endcsname{.notdef.031}%
947     \expandafter\def\csname krntst@T1@156\endcsname{.notdef.156}%
948     \expandafter\def\csname krntst@T1@188\endcsname{.notdef.188}%
949   \else
950     \expandafter\def\csname krntst@T1@027\endcsname{ff}%
951     \expandafter\def\csname krntst@T1@028\endcsname{fi}%
952     \expandafter\def\csname krntst@T1@029\endcsname{fl}%
953     \expandafter\def\csname krntst@T1@030\endcsname{ffi}%
954     \expandafter\def\csname krntst@T1@031\endcsname{ffl}%
955     \expandafter\def\csname krntst@T1@156\endcsname{IJ}%
956     \expandafter\def\csname krntst@T1@188\endcsname{ij}%
957   \fi
958   \ifnum#1<-1\relax
959     \expandafter\def\csname krntst@T1@073\endcsname{.notdef.073}%
960     \expandafter\def\csname krntst@T1@099\endcsname{.notdef.099}%
961     \expandafter\def\csname krntst@T1@102\endcsname{.notdef.102}%
962     \expandafter\def\csname krntst@T1@105\endcsname{.notdef.105}%

```

```

963   \expandafter\def\csname krntst@T1@115\endcsname{.notdef.115}%
964   \else
965     \expandafter\def\csname krntst@T1@073\endcsname{I}%
966     \expandafter\def\csname krntst@T1@099\endcsname{c}%
967     \expandafter\def\csname krntst@T1@102\endcsname{f}%
968     \expandafter\def\csname krntst@T1@105\endcsname{i}%
969     \expandafter\def\csname krntst@T1@115\endcsname{s}%
970   \fi
971 }
972 \typeout{^^JValid values for T1 encoding:}%
973 \typeout{ligaturing: -2, -1, 0, 1}%
974 \typeout{Defaults for T1 encoding:}%
975 \setkeys{krnenc}{ligaturing=1}%
976 \typeout{}%
977 \makeatother

Now, set \getpsname.
978 \newcommand\getpsname[1]{%
979   \ifcase#1%
980     grave\or% 000
981     acute\or% 001
982     circumflex\or% 002
983     tilde\or% 003
984     dieresis\or% 004
985     hungarumlaut\or% 005
986     ring\or% 006
987     caron\or% 007
988     breve\or% 008
989     macron\or% 009
990     dotaccent\or% 010
991     cedilla\or% 011
992     ogonek\or% 012
993     quotesinglbase\or% 013
994     guilsinglleft\or% 014
995     guilsinglright\or% 015
996     quotedblleft\or% 016
997     quotedblright\or% 017
998     quotedblbase\or% 018
999     guillemotleft\or% 019
1000    guillemotright\or% 020
1001    rangedash\or% 021
1002    punctdash\or% 022
1003    compwordmark\or% 023
1004    perthousandzero\or% 024
1005    dotlessi\or% 025
1006    dotlessj\or% 026
1007    \csname krntst@T1@027\endcsname\or % 027
1008    \csname krntst@T1@028\endcsname\or % 028
1009    \csname krntst@T1@029\endcsname\or % 029
1010    \csname krntst@T1@030\endcsname\or % 030
1011    \csname krntst@T1@031\endcsname\or % 031
1012    visiblespace\or% 032
1013    exclam\or% 033
1014    quotedbl\or% 034
1015    numbersign\or% 035

```

1016 dollar\or% 036  
1017 percent\or% 037  
1018 ampersand\or% 038  
1019 quoteright\or% 039  
1020 parenleft\or% 040  
1021 parenright\or% 041  
1022 asterisk\or% 042  
1023 plus\or% 043  
1024 comma\or% 044  
1025 hyphen\or% 045  
1026 period\or% 046  
1027 slash\or% 047  
1028 zero\or% 048  
1029 one\or% 049  
1030 two\or% 050  
1031 three\or% 051  
1032 four\or% 052  
1033 five\or% 053  
1034 six\or% 054  
1035 seven\or% 055  
1036 eight\or% 056  
1037 nine\or% 057  
1038 colon\or% 058  
1039 semicolon\or% 059  
1040 less\or% 060  
1041 equal\or% 061  
1042 greater\or% 062  
1043 question\or% 063  
1044 at\or% 064  
1045 A\or% 065  
1046 B\or% 066  
1047 C\or% 067  
1048 D\or% 068  
1049 E\or% 069  
1050 F\or% 070  
1051 G\or% 071  
1052 H\or% 072  
1053 \csname krontst@T1@073\endcsname\or % 073  
1054 J\or% 074  
1055 K\or% 075  
1056 L\or% 076  
1057 M\or% 077  
1058 N\or% 078  
1059 O\or% 079  
1060 P\or% 080  
1061 Q\or% 081  
1062 R\or% 082  
1063 S\or% 083  
1064 T\or% 084  
1065 U\or% 085  
1066 V\or% 086  
1067 W\or% 087  
1068 X\or% 088  
1069 Y\or% 089

1070 Z\or% 090  
1071 bracketleft\or% 091  
1072 backslash\or% 092  
1073 bracketright\or% 093  
1074 asciicircum\or% 094  
1075 underscore\or% 095  
1076 quoteleft\or% 096  
1077 a\or% 097  
1078 b\or% 098  
1079 \csname krntst@T1@099\endcsname\or % 099  
1080 d\or% 100  
1081 e\or% 101  
1082 \csname krntst@T1@102\endcsname\or % 102  
1083 g\or% 103  
1084 h\or% 104  
1085 \csname krntst@T1@105\endcsname\or % 105  
1086 j\or% 106  
1087 k\or% 107  
1088 l\or% 108  
1089 m\or% 109  
1090 n\or% 110  
1091 o\or% 111  
1092 p\or% 112  
1093 q\or% 113  
1094 r\or% 114  
1095 \csname krntst@T1@115\endcsname\or % 115  
1096 t\or% 116  
1097 u\or% 117  
1098 v\or% 118  
1099 w\or% 119  
1100 x\or% 120  
1101 y\or% 121  
1102 z\or% 122  
1103 braceleft\or% 123  
1104 bar\or% 124  
1105 braceright\or% 125  
1106 asciitilde\or% 126  
1107 hyphenchar\or% 127  
1108 Abreve\or% 128  
1109 Aogonek\or% 129  
1110 Cacute\or% 130  
1111 Ccaron\or% 131  
1112 Dcaron\or% 132  
1113 Ecaron\or% 133  
1114 Eogonek\or% 134  
1115 Gbreve\or% 135  
1116 Lacute\or% 136  
1117 Lcaron\or% 137  
1118 Lslash\or% 138  
1119 Nacute\or% 139  
1120 Ncaron\or% 140  
1121 Ng\or% 141  
1122 Ohungarumlaut\or% 142  
1123 Racute\or% 143

1124 Rcaron\or% 144  
1125 Sacute\or% 145  
1126 Scaron\or% 146  
1127 Scedilla\or% 147  
1128 Tcaron\or% 148  
1129 Tcedilla\or% 149  
1130 Uhungarumlaut\or% 150  
1131 Uring\or% 151  
1132 Ydieresis\or% 152  
1133 Zacute\or% 153  
1134 Zcaron\or% 154  
1135 Zdotaccent\or% 155  
1136 \csname krntst@T1@156\endcsname\or % 156  
1137 Idotaccent\or% 157  
1138 dbar\or% 158  
1139 section\or% 159  
1140 abreve\or% 160  
1141 aogonek\or% 161  
1142 cacute\or% 162  
1143 ccaron\or% 163  
1144 dcaron\or% 164  
1145 ecaron\or% 165  
1146 eogonek\or% 166  
1147 gbreve\or% 167  
1148 lacute\or% 168  
1149 lcaron\or% 169  
1150 lslash\or% 170  
1151 nacute\or% 171  
1152 ncaron\or% 172  
1153 ng\or% 173  
1154 ohungarumlaut\or% 174  
1155 racute\or% 175  
1156 rcaron\or% 176  
1157 sacute\or% 177  
1158 scaron\or% 178  
1159 scedilla\or% 179  
1160 tcaron\or% 180  
1161 tcedilla\or% 181  
1162 uhungarumlaut\or% 182  
1163 uring\or% 183  
1164 ydieresis\or% 184  
1165 zacute\or% 185  
1166 zcaron\or% 186  
1167 zdotaccent\or% 187  
1168 \csname krntst@T1@188\endcsname\or % 188  
1169 exclamdown\or% 189  
1170 questiondown\or% 190  
1171 sterling\or% 191  
1172 Agrave\or% 192  
1173 Aacute\or% 193  
1174 Acircumflex\or% 194  
1175 Atilde\or% 195  
1176 Adieresis\or% 196  
1177 Aring\or% 197

1178 AE\or% 198  
1179 Ccedilla\or% 199  
1180 Egrave\or% 200  
1181 Eacute\or% 201  
1182 Ecircumflex\or% 202  
1183 Edieresis\or% 203  
1184 Igrave\or% 204  
1185 Iacute\or% 205  
1186 Icircumflex\or% 206  
1187 Idieresis\or% 207  
1188 Eth\or% 208  
1189 Ntilde\or% 209  
1190 Ograve\or% 210  
1191 Oacute\or% 211  
1192 Ocircumflex\or% 212  
1193 Otilde\or% 213  
1194 Odieresis\or% 214  
1195 OE\or% 215  
1196 Oslash\or% 216  
1197 Ugrave\or% 217  
1198 Uacute\or% 218  
1199 Ucircumflex\or% 219  
1200 Udieresis\or% 220  
1201 Yacute\or% 221  
1202 Thorn\or% 222  
1203 SS\or% 223  
1204 agrave\or% 224  
1205 aacute\or% 225  
1206 acircumflex\or% 226  
1207 atilde\or% 227  
1208 adieresis\or% 228  
1209 aring\or% 229  
1210 ae\or% 230  
1211 ccedilla\or% 231  
1212 egrave\or% 232  
1213 eacute\or% 233  
1214 ecircumflex\or% 234  
1215 edieresis\or% 235  
1216 igrave\or% 236  
1217 iacute\or% 237  
1218 icircumflex\or% 238  
1219 idieresis\or% 239  
1220 eth\or% 240  
1221 ntilde\or% 241  
1222 ograve\or% 242  
1223 oacute\or% 243  
1224 ocircumflex\or% 244  
1225 otilde\or% 245  
1226 odieresis\or% 246  
1227 oe\or% 247  
1228 oslash\or% 248  
1229 ugrave\or% 249  
1230 uacute\or% 250  
1231 ucircumflex\or% 251

```

1232 udieresis\or% 252
1233 yacute\or% 253
1234 thorn\or% 254
1235 germandbls\fi% 255
1236 }
1237 </mtx & t1>

```

### 6.3.2 TS1 encoding

The TS1 encoding. The data are taken from `ts1.etx`.

For unknown slots, a strange Postscript name is returned, but no warning is generated.

```

1238 (*mtx & ts1)
1239 \newcommand\getpsname[1]{%
1240   \ifcase#1%
1241     capitalgrave\or% 000
1242     capitalacute\or% 001
1243     capitalcircumflex\or% 002
1244     capitalthilde\or% 003
1245     capitaldieresis\or% 004
1246     capitalhungarumlaut\or% 005
1247     capitalring\or% 006
1248     capitalcaron\or% 007
1249     capitalbreve\or% 008
1250     capitalmacron\or% 009
1251     capitaldotaccent\or% 010
1252     cedilla\or% 011
1253     ogonek\or% 012
1254     quotesinglbase\or% 013
1255     .notdef.014\or% 014
1256     .notdef.015\or% 015
1257     .notdef.016\or% 016
1258     .notdef.017\or% 017
1259     quotedblbase\or% 018
1260     .notdef.019\or% 019
1261     .notdef.020\or% 020
1262     twelveudash\or% 021
1263     threequartersemdash\or% 022
1264     capitalcompwordmark\or% 023
1265     arrowleft\or% 024
1266     arrowright\or% 025
1267     tieaccentlowercase\or% 026
1268     tieaccentcapital\or% 027
1269     newtieaccentlowercase\or% 028
1270     newtieaccentcapital\or% 029
1271     ascendercompwordmark\or% 030
1272     blank\or% 031
1273     .notdef.032\or% 032
1274     .notdef.033\or% 033
1275     .notdef.034\or% 034
1276     .notdef.035\or% 035
1277     dollar\or% 036
1278     .notdef.037\or% 037
1279     .notdef.038\or% 038

```

1280 quotesingle\or% 039  
1281 .notdef.040\or% 040  
1282 .notdef.041\or% 041  
1283 asteriskcentered\or% 042  
1284 .notdef.043\or% 043  
1285 comma\or% 044  
1286 hyphendbl\or% 045  
1287 period\or% 046  
1288 fraction\or% 047  
1289 zerooldstyle\or% 048  
1290 oneoldstyle\or% 049  
1291 twooldstyle\or% 050  
1292 threeoldstyle\or% 051  
1293 fouroldstyle\or% 052  
1294 fiveoldstyle\or% 053  
1295 sixoldstyle\or% 054  
1296 sevenoldstyle\or% 055  
1297 eightoldstyle\or% 056  
1298 nineoldstyle\or% 057  
1299 .notdef.058\or% 058  
1300 .notdef.059\or% 059  
1301 angbracketleft\or% 060  
1302 minus\or% 061  
1303 angbracketright\or% 062  
1304 .notdef.063\or% 063  
1305 .notdef.064\or% 064  
1306 .notdef.065\or% 065  
1307 .notdef.066\or% 066  
1308 .notdef.067\or% 067  
1309 .notdef.068\or% 068  
1310 .notdef.069\or% 069  
1311 .notdef.070\or% 070  
1312 .notdef.071\or% 071  
1313 .notdef.072\or% 072  
1314 .notdef.073\or% 073  
1315 .notdef.074\or% 074  
1316 .notdef.075\or% 075  
1317 .notdef.076\or% 076  
1318 Omegainv\or% 077  
1319 .notdef.078\or% 078  
1320 bigcircle\or% 079  
1321 .notdef.080\or% 080  
1322 .notdef.081\or% 081  
1323 .notdef.082\or% 082  
1324 .notdef.083\or% 083  
1325 .notdef.084\or% 084  
1326 .notdef.085\or% 085  
1327 .notdef.086\or% 086  
1328 Omega\or% 087  
1329 .notdef.088\or% 088  
1330 .notdef.089\or% 089  
1331 .notdef.090\or% 090  
1332 openbracketleft\or% 091  
1333 .notdef.092\or% 092

1334 openbracketright\or% 093  
1335 arrowup\or% 094  
1336 arrowdown\or% 095  
1337 asciigrave\or% 096  
1338 .notdef.097\or% 097  
1339 born\or% 098  
1340 divorced\or% 099  
1341 died\or% 100  
1342 .notdef.101\or% 101  
1343 .notdef.102\or% 102  
1344 .notdef.103\or% 103  
1345 .notdef.104\or% 104  
1346 .notdef.105\or% 105  
1347 .notdef.106\or% 106  
1348 .notdef.107\or% 107  
1349 leaf\or% 108  
1350 married\or% 109  
1351 musicalnote\or% 110  
1352 .notdef.111\or% 111  
1353 .notdef.112\or% 112  
1354 .notdef.113\or% 113  
1355 .notdef.114\or% 114  
1356 .notdef.115\or% 115  
1357 .notdef.116\or% 116  
1358 .notdef.117\or% 117  
1359 .notdef.118\or% 118  
1360 .notdef.119\or% 119  
1361 .notdef.120\or% 120  
1362 .notdef.121\or% 121  
1363 .notdef.122\or% 122  
1364 .notdef.123\or% 123  
1365 .notdef.124\or% 124  
1366 .notdef.125\or% 125  
1367 tildelow\or% 126  
1368 hyphendblchar\or% 127  
1369 asciibreve\or% 128  
1370 asciicaron\or% 129  
1371 asciiacutedbl\or% 130  
1372 asciigravedbl\or% 131  
1373 dagger\or% 132  
1374 daggerdbl\or% 133  
1375 bardbl\or% 134  
1376 perthousand\or% 135  
1377 bullet\or% 136  
1378 centigrade\or% 137  
1379 dollaroldstyle\or% 138  
1380 centoldstyle\or% 139  
1381 florin\or% 140  
1382 colonmonetary\or% 141  
1383 won\or% 142  
1384 naira\or% 143  
1385 guarani\or% 144  
1386 peso\or% 145  
1387 lira\or% 146

1388 recipe\or% 147  
1389 interrobang\or% 148  
1390 interrobangdown\or% 149  
1391 dong\or% 150  
1392 trademark\or% 151  
1393 pertenthousand\or% 152  
1394 pilcrow\or% 153  
1395 baht\or% 154  
1396 numero\or% 155  
1397 discount\or% 156  
1398 estimated\or% 157  
1399 openbullet\or% 158  
1400 servicemark\or% 159  
1401 quillbracketleft\or% 160  
1402 quillbracketright\or% 161  
1403 cent\or% 162  
1404 sterling\or% 163  
1405 currency\or% 164  
1406 yen\or% 165  
1407 brokenbar\or% 166  
1408 section\or% 167  
1409 asciidieresis\or% 168  
1410 copyright\or% 169  
1411 ordfeminine\or% 170  
1412 copyleft\or% 171  
1413 logicalnot\or% 172  
1414 circledP\or% 173  
1415 registered\or% 174  
1416 asciimacron\or% 175  
1417 degree\or% 176  
1418 plusminus\or% 177  
1419 twosuperior\or% 178  
1420 threesuperior\or% 179  
1421 asciiacute\or% 180  
1422 mu\or% 181  
1423 paragraph\or% 182  
1424 periodcentered\or% 183  
1425 referencemark\or% 184  
1426 onesuperior\or% 185  
1427 ordmasculine\or% 186  
1428 radical\or% 187  
1429 onequarter\or% 188  
1430 onehalf\or% 189  
1431 threequarters\or% 190  
1432 euro\or% 191  
1433 .notdef.192\or% 192  
1434 .notdef.193\or% 193  
1435 .notdef.194\or% 194  
1436 .notdef.195\or% 195  
1437 .notdef.196\or% 196  
1438 .notdef.197\or% 197  
1439 .notdef.198\or% 198  
1440 .notdef.199\or% 199  
1441 .notdef.200\or% 200

1442 .notdef.201\or% 201  
1443 .notdef.202\or% 202  
1444 .notdef.203\or% 203  
1445 .notdef.204\or% 204  
1446 .notdef.205\or% 205  
1447 .notdef.206\or% 206  
1448 .notdef.207\or% 207  
1449 .notdef.208\or% 208  
1450 .notdef.209\or% 209  
1451 .notdef.210\or% 210  
1452 .notdef.211\or% 211  
1453 .notdef.212\or% 212  
1454 .notdef.213\or% 213  
1455 multiply\or% 214  
1456 .notdef.215\or% 215  
1457 .notdef.216\or% 216  
1458 .notdef.217\or% 217  
1459 .notdef.218\or% 218  
1460 .notdef.219\or% 219  
1461 .notdef.220\or% 220  
1462 .notdef.221\or% 221  
1463 .notdef.222\or% 222  
1464 .notdef.223\or% 223  
1465 .notdef.224\or% 224  
1466 .notdef.225\or% 225  
1467 .notdef.226\or% 226  
1468 .notdef.227\or% 227  
1469 .notdef.228\or% 228  
1470 .notdef.229\or% 229  
1471 .notdef.230\or% 230  
1472 .notdef.231\or% 231  
1473 .notdef.232\or% 232  
1474 .notdef.233\or% 233  
1475 .notdef.234\or% 234  
1476 .notdef.235\or% 235  
1477 .notdef.236\or% 236  
1478 .notdef.237\or% 237  
1479 .notdef.238\or% 238  
1480 .notdef.239\or% 239  
1481 .notdef.240\or% 240  
1482 .notdef.241\or% 241  
1483 .notdef.242\or% 242  
1484 .notdef.243\or% 243  
1485 .notdef.244\or% 244  
1486 .notdef.245\or% 245  
1487 divide\or% 246  
1488 .notdef.247\or% 247  
1489 .notdef.248\or% 248  
1490 .notdef.249\or% 249  
1491 .notdef.250\or% 250  
1492 .notdef.251\or% 251  
1493 .notdef.252\or% 252  
1494 .notdef.253\or% 253  
1495 .notdef.254\or% 254

```

1496 .notdef.255\fi% 255
1497 }
1498 </mtx & ts1>

```

### 6.3.3 OT1 encoding

The OT1 encoding. The data are taken from `ot1.etx`.

For unknown slots, a strange Postscript name is returned, but no warning is generated.

The OT1 encoding uses different font names depending on upright/italic and ligatures. This is handled using the `\OT1@{slot}` commands that are set for an upright roman font by default.

Set the default glyph names.

```

1499 <*mtx & ot1>
1500 \makeatletter

Set options to switch to other font shapes.
1501 \define@key{krnenc}{ligaturing}[2]{%
1502   \typeout{OT1 encoding: ligaturing = #1}%
1503   \ifnum#1=2\relax
1504     \expandafter\def\csname krntst@OT1@011\endcsname{ff}%
1505     \expandafter\def\csname krntst@OT1@012\endcsname{fi}%
1506     \expandafter\def\csname krntst@OT1@013\endcsname{fl}%
1507     \expandafter\def\csname krntst@OT1@014\endcsname{ffi}%
1508     \expandafter\def\csname krntst@OT1@015\endcsname{ffl}%
1509     \expandafter\def\csname krntst@OT1@060\endcsname{exclamdown}%
1510     \expandafter\def\csname krntst@OT1@062\endcsname{questiondown}%
1511   \else
1512     \expandafter\def\csname krntst@OT1@011\endcsname{arrowup}%
1513     \expandafter\def\csname krntst@OT1@012\endcsname{arrowdown}%
1514     \expandafter\def\csname krntst@OT1@013\endcsname{quotesingle}%
1515     \expandafter\def\csname krntst@OT1@014\endcsname{exclamdown}%
1516     \expandafter\def\csname krntst@OT1@015\endcsname{questiondown}%
1517     \expandafter\def\csname krntst@OT1@060\endcsname{less}%
1518     \expandafter\def\csname krntst@OT1@062\endcsname{greater}%
1519   \fi
1520   \ifnum#1=0\relax
1521     \expandafter\def\csname krntst@OT1@032\endcsname{visibleospace}%
1522     \expandafter\def\csname krntst@OT1@034\endcsname{quotedbl}%
1523     \expandafter\def\csname krntst@OT1@092\endcsname{backslash}%
1524     \expandafter\def\csname krntst@OT1@095\endcsname{underscore}%
1525     \expandafter\def\csname krntst@OT1@123\endcsname{braceleft}%
1526     \expandafter\def\csname krntst@OT1@124\endcsname{bar}%
1527     \expandafter\def\csname krntst@OT1@125\endcsname{braceright}%
1528   \else
1529     \expandafter\def\csname krntst@OT1@032\endcsname{lslash}%
1530     \expandafter\def\csname krntst@OT1@034\endcsname{quotedblright}%
1531     \expandafter\def\csname krntst@OT1@092\endcsname{quotedblleft}%
1532     \expandafter\def\csname krntst@OT1@095\endcsname{dotaccent}%
1533     \expandafter\def\csname krntst@OT1@123\endcsname{rangedash}%
1534     \expandafter\def\csname krntst@OT1@124\endcsname{punctdash}%
1535     \expandafter\def\csname krntst@OT1@125\endcsname{hungarumlaut}%
1536   \fi
1537 }

```

```

1538 \define@key{krnenc}{italicizing}[true]{%
1539 \typeout{OT1 encoding: italicizing = #1}%
1540 \csname if#1\endcsname
1541 \expandafter\def\csname krntst@OT1@036\endcsname{sterling}%
1542 \else
1543 \expandafter\def\csname krntst@OT1@036\endcsname{dollar}
1544 \fi
1545 }
1546 \typeout{^^JValid values for OT1 encoding:}%
1547 \typeout{ligaturing: 0, 1, 2}%
1548 \typeout{italicizing: false, true}%
1549 \typeout{Defaults for OT1 encoding:}%
1550 \setkeys{krnenc}{ligaturing=2,italicizing=false}%
1551 \typeout{}%
1552 \makeatother

Now, set \getpsname.
1553 \newcommand\getpsname[1]{%
1554 \ifcase#1%
1555 Gamma\or % 000
1556 Delta\or % 001
1557 Theta\or % 002
1558 Lambda\or % 003
1559 Xi\or % 004
1560 Pi\or % 005
1561 Sigma\or % 006
1562 Upsilon\or % 007
1563 Phi\or % 008
1564 Psi\or % 009
1565 Omega\or % 010
1566 \csname krntst@OT1@011\endcsname\or % 011
1567 \csname krntst@OT1@012\endcsname\or % 012
1568 \csname krntst@OT1@013\endcsname\or % 013
1569 \csname krntst@OT1@014\endcsname\or % 014
1570 \csname krntst@OT1@015\endcsname\or % 015
1571 dotlessi\or % 016
1572 dotlessj\or % 017
1573 grave\or % 018
1574 acute\or % 019
1575 caron\or % 020
1576 breve\or % 021
1577 macron\or % 022
1578 ringfitted\or % 023
1579 cedilla\or % 024
1580 germandbls\or % 025
1581 ae\or % 026
1582 oe\or % 027
1583 oslash\or % 028
1584 AE\or % 029
1585 OE\or % 030
1586 Oslash\or % 031
1587 \csname krntst@OT1@032\endcsname\or % 032
1588 exclam\or % 033
1589 \csname krntst@OT1@034\endcsname\or % 034
1590 numbersign\or % 035

```

1591 \csname krtst@OT1@036\endcsname\or % 036  
1592 percent\or % 037  
1593 ampersand\or % 038  
1594 quoteright\or % 039  
1595 parenleft\or % 040  
1596 parenright\or % 041  
1597 asterisk\or % 042  
1598 plus\or % 043  
1599 comma\or % 044  
1600 hyphen\or % 045  
1601 period\or % 046  
1602 slash\or % 047  
1603 zero\or % 048  
1604 one\or % 049  
1605 two\or % 050  
1606 three\or % 051  
1607 four\or % 052  
1608 five\or % 053  
1609 six\or % 054  
1610 seven\or % 055  
1611 eight\or % 056  
1612 nine\or % 057  
1613 colon\or % 058  
1614 semicolon\or % 059  
1615 \csname krtst@OT1@060\endcsname\or % 060  
1616 equal\or % 061  
1617 \csname krtst@OT1@062\endcsname\or % 062  
1618 question\or % 063  
1619 at\or % 064  
1620 A\or % 065  
1621 B\or % 066  
1622 C\or % 067  
1623 D\or % 068  
1624 E\or % 069  
1625 F\or % 070  
1626 G\or % 071  
1627 H\or % 072  
1628 I\or % 073  
1629 J\or % 074  
1630 K\or % 075  
1631 L\or % 076  
1632 M\or % 077  
1633 N\or % 078  
1634 O\or % 079  
1635 P\or % 080  
1636 Q\or % 081  
1637 R\or % 082  
1638 S\or % 083  
1639 T\or % 084  
1640 U\or % 085  
1641 V\or % 086  
1642 W\or % 087  
1643 X\or % 088  
1644 Y\or % 089

1645 Z\or % 090  
1646 bracketleft\or % 091  
1647 \csname krntst@OT1@092\endcsname\or % 092  
1648 bracketright\or % 093  
1649 circumflex\or % 094  
1650 \csname krntst@OT1@095\endcsname\or % 095  
1651 quoteleft\or % 096  
1652 a\or % 097  
1653 b\or % 098  
1654 c\or % 099  
1655 d\or % 100  
1656 e\or % 101  
1657 f\or % 102  
1658 g\or % 103  
1659 h\or % 104  
1660 i\or % 105  
1661 j\or % 106  
1662 k\or % 107  
1663 l\or % 108  
1664 m\or % 109  
1665 n\or % 110  
1666 o\or % 111  
1667 p\or % 112  
1668 q\or % 113  
1669 r\or % 114  
1670 s\or % 115  
1671 t\or % 116  
1672 u\or % 117  
1673 v\or % 118  
1674 w\or % 119  
1675 x\or % 120  
1676 y\or % 121  
1677 z\or % 122  
1678 \csname krntst@OT1@123\endcsname\or % 123  
1679 \csname krntst@OT1@124\endcsname\or % 124  
1680 \csname krntst@OT1@125\endcsname\or % 125  
1681 tilde\or % 126  
1682 dieresis\or % 127  
1683 .notdef.128\or % 128  
1684 .notdef.129\or % 129  
1685 .notdef.130\or % 130  
1686 .notdef.131\or % 131  
1687 .notdef.132\or % 132  
1688 .notdef.133\or % 133  
1689 .notdef.134\or % 134  
1690 .notdef.135\or % 135  
1691 .notdef.136\or % 136  
1692 .notdef.137\or % 137  
1693 Lslash\or % 138  
1694 .notdef.139\or % 139  
1695 .notdef.140\or % 140  
1696 .notdef.141\or % 141  
1697 .notdef.142\or % 142  
1698 .notdef.143\or % 143

1699 .notdef.144\or % 144  
1700 .notdef.145\or % 145  
1701 .notdef.146\or % 146  
1702 .notdef.147\or % 147  
1703 .notdef.148\or % 148  
1704 .notdef.149\or % 149  
1705 .notdef.150\or % 150  
1706 .notdef.151\or % 151  
1707 .notdef.152\or % 152  
1708 .notdef.153\or % 153  
1709 .notdef.154\or % 154  
1710 .notdef.155\or % 155  
1711 .notdef.156\or % 156  
1712 .notdef.157\or % 157  
1713 .notdef.158\or % 158  
1714 .notdef.159\or % 159  
1715 .notdef.160\or % 160  
1716 .notdef.161\or % 161  
1717 .notdef.162\or % 162  
1718 .notdef.163\or % 163  
1719 .notdef.164\or % 164  
1720 .notdef.165\or % 165  
1721 .notdef.166\or % 166  
1722 .notdef.167\or % 167  
1723 .notdef.168\or % 168  
1724 .notdef.169\or % 169  
1725 lslash\or % 170  
1726 .notdef.171\or % 171  
1727 .notdef.172\or % 172  
1728 .notdef.173\or % 173  
1729 .notdef.174\or % 174  
1730 .notdef.175\or % 175  
1731 .notdef.176\or % 176  
1732 .notdef.177\or % 177  
1733 .notdef.178\or % 178  
1734 .notdef.179\or % 179  
1735 .notdef.180\or % 180  
1736 .notdef.181\or % 181  
1737 .notdef.182\or % 182  
1738 .notdef.183\or % 183  
1739 .notdef.184\or % 184  
1740 .notdef.185\or % 185  
1741 .notdef.186\or % 186  
1742 .notdef.187\or % 187  
1743 .notdef.188\or % 188  
1744 .notdef.189\or % 189  
1745 .notdef.190\or % 190  
1746 .notdef.191\or % 191  
1747 .notdef.192\or % 192  
1748 .notdef.193\or % 193  
1749 .notdef.194\or % 194  
1750 .notdef.195\or % 195  
1751 .notdef.196\or % 196  
1752 .notdef.197\or % 197

1753 .notdef.198\or % 198  
1754 .notdef.199\or % 199  
1755 .notdef.200\or % 200  
1756 .notdef.201\or % 201  
1757 .notdef.202\or % 202  
1758 .notdef.203\or % 203  
1759 .notdef.204\or % 204  
1760 .notdef.205\or % 205  
1761 .notdef.206\or % 206  
1762 .notdef.207\or % 207  
1763 .notdef.208\or % 208  
1764 .notdef.209\or % 209  
1765 .notdef.210\or % 210  
1766 .notdef.211\or % 211  
1767 .notdef.212\or % 212  
1768 .notdef.213\or % 213  
1769 .notdef.214\or % 214  
1770 .notdef.215\or % 215  
1771 .notdef.216\or % 216  
1772 .notdef.217\or % 217  
1773 .notdef.218\or % 218  
1774 .notdef.219\or % 219  
1775 .notdef.220\or % 220  
1776 .notdef.221\or % 221  
1777 .notdef.222\or % 222  
1778 .notdef.223\or % 223  
1779 .notdef.224\or % 224  
1780 .notdef.225\or % 225  
1781 .notdef.226\or % 226  
1782 .notdef.227\or % 227  
1783 .notdef.228\or % 228  
1784 .notdef.229\or % 229  
1785 .notdef.230\or % 230  
1786 .notdef.231\or % 231  
1787 .notdef.232\or % 232  
1788 .notdef.233\or % 233  
1789 .notdef.234\or % 234  
1790 .notdef.235\or % 235  
1791 .notdef.236\or % 236  
1792 .notdef.237\or % 237  
1793 .notdef.238\or % 238  
1794 .notdef.239\or % 239  
1795 .notdef.240\or % 240  
1796 .notdef.241\or % 241  
1797 .notdef.242\or % 242  
1798 .notdef.243\or % 243  
1799 .notdef.244\or % 244  
1800 .notdef.245\or % 245  
1801 .notdef.246\or % 246  
1802 .notdef.247\or % 247  
1803 .notdef.248\or % 248  
1804 .notdef.249\or % 249  
1805 .notdef.250\or % 250  
1806 .notdef.251\or % 251

```

1807 .notdef.252\or % 252
1808 .notdef.253\or % 253
1809 .notdef.254\or % 254
1810 .notdef.255\fi % 255
1811 }
1812 </mtx & ot1>

```

### 6.3.4 T2A encoding

The T2A encoding. The data are taken from `t2a.etx`.

For unknown slots, a strange Postscript name is returned, but no warning is generated.

```

1813 (*mtx & t2a)
1814 \newcommand\getpsname[1]{%
1815   \ifcase#1%
1816   grave\or% 000
1817   acute\or% 001
1818   circumflex\or% 002
1819   tilde\or% 003
1820   dieresis\or% 004
1821   hungarumlaut\or% 005
1822   ring\or% 006
1823   caron\or% 007
1824   breve\or% 008
1825   macron\or% 009
1826   dotaccent\or% 010
1827   cedilla\or% 011
1828   ogonek\or% 012
1829   CYRpalochka\or% 013
1830   angleleft\or% 014
1831   angleright\or% 015
1832   quotedblleft\or% 016
1833   quotedblright\or% 017
1834   cyrflex\or% 018
1835   dblgrave\or% 019
1836   cyrbreve\or% 020
1837   rangedash\or% 021
1838   cyrdash\or% 022
1839   compwordmark\or% 023
1840   perthousandzero\or% 024
1841   dotlessi\or% 025
1842   dotlessj\or% 026
1843   ff\or% 027
1844   fi\or% 028
1845   fl\or% 029
1846   ffi\or% 030
1847   ffl\or% 031
1848   visiblespace\or% 032
1849   exclam\or% 033
1850   quotedbl\or% 034
1851   numbersign\or% 035
1852   dollar\or% 036
1853   percent\or% 037
1854   ampersand\or% 038

```

1855 quoteright\or% 039  
1856 parenleft\or% 040  
1857 parenright\or% 041  
1858 asterisk\or% 042  
1859 plus\or% 043  
1860 comma\or% 044  
1861 hyphen\or% 045  
1862 period\or% 046  
1863 slash\or% 047  
1864 zero\or% 048  
1865 one\or% 049  
1866 two\or% 050  
1867 three\or% 051  
1868 four\or% 052  
1869 five\or% 053  
1870 six\or% 054  
1871 seven\or% 055  
1872 eight\or% 056  
1873 nine\or% 057  
1874 colon\or% 058  
1875 semicolon\or% 059  
1876 less\or% 060  
1877 equal\or% 061  
1878 greater\or% 062  
1879 question\or% 063  
1880 at\or% 064  
1881 A\or% 065  
1882 B\or% 066  
1883 C\or% 067  
1884 D\or% 068  
1885 E\or% 069  
1886 F\or% 070  
1887 G\or% 071  
1888 H\or% 072  
1889 I\or% 073  
1890 J\or% 074  
1891 K\or% 075  
1892 L\or% 076  
1893 M\or% 077  
1894 N\or% 078  
1895 O\or% 079  
1896 P\or% 080  
1897 Q\or% 081  
1898 R\or% 082  
1899 S\or% 083  
1900 T\or% 084  
1901 U\or% 085  
1902 V\or% 086  
1903 W\or% 087  
1904 X\or% 088  
1905 Y\or% 089  
1906 Z\or% 090  
1907 bracketleft\or% 091  
1908 backslash\or% 092

1909 bracketright\or% 093  
1910 asciicircum\or% 094  
1911 underscore\or% 095  
1912 quoteleft\or% 096  
1913 a\or% 097  
1914 b\or% 098  
1915 c\or% 099  
1916 d\or% 100  
1917 e\or% 101  
1918 f\or% 102  
1919 g\or% 103  
1920 h\or% 104  
1921 i\or% 105  
1922 j\or% 106  
1923 k\or% 107  
1924 l\or% 108  
1925 m\or% 109  
1926 n\or% 110  
1927 o\or% 111  
1928 p\or% 112  
1929 q\or% 113  
1930 r\or% 114  
1931 s\or% 115  
1932 t\or% 116  
1933 u\or% 117  
1934 v\or% 118  
1935 w\or% 119  
1936 x\or% 120  
1937 y\or% 121  
1938 z\or% 122  
1939 braceleft\or% 123  
1940 bar\or% 124  
1941 braceright\or% 125  
1942 asciitilde\or% 126  
1943 hyphenchar\or% 127  
1944 CYRGUP\or% 128  
1945 CYRGHCRS\or% 129  
1946 CYRDJE\or% 130  
1947 CYRTSHE\or% 131  
1948 CYRSHHA\or% 132  
1949 CYRZHDSC\or% 133  
1950 CYRZDSC\or% 134  
1951 CYRLJE\or% 135  
1952 CYRYI\or% 136  
1953 CYRKDSC\or% 137  
1954 CYRKBEAK\or% 138  
1955 CYRKVCRS\or% 139  
1956 CYRAE\or% 140  
1957 CYRNDSC\or% 141  
1958 CYRNG\or% 142  
1959 CYRDZE\or% 143  
1960 CYROTLD\or% 144  
1961 CYRSDSC\or% 145  
1962 CYRUSHRT\or% 146

1963 CYRY\or% 147  
1964 CYRYHCRS\or% 148  
1965 CYRHDSC\or% 149  
1966 CYRDZHE\or% 150  
1967 CYRCHVCRS\or% 151  
1968 CYRCHRDC\or% 152  
1969 CYRIE\or% 153  
1970 CYRSCHWA\or% 154  
1971 CYRNJE\or% 155  
1972 CYRYO\or% 156  
1973 numero\or% 157  
1974 currency\or% 158  
1975 section\or% 159  
1976 cyrgup\or% 160  
1977 cyrghcrs\or% 161  
1978 cyrdje\or% 162  
1979 cyrtshe\or% 163  
1980 cyrshha\or% 164  
1981 cyrzhdsc\or% 165  
1982 cyrzdsc\or% 166  
1983 cyrlje\or% 167  
1984 cyryi\or% 168  
1985 cyrkdsc\or% 169  
1986 cyrkbeak\or% 170  
1987 cyrkvcrs\or% 171  
1988 cyrae\or% 172  
1989 cyrndsc\or% 173  
1990 cyrng\or% 174  
1991 cyrdze\or% 175  
1992 cyrotld\or% 176  
1993 cyrsdsc\or% 177  
1994 cyrushrt\or% 178  
1995 cyry\or% 179  
1996 cyryhcrs\or% 180  
1997 cyrhdsc\or% 181  
1998 cyrdzhe\or% 182  
1999 cyrchvcrs\or% 183  
2000 cyrchrsc\or% 184  
2001 cyrie\or% 185  
2002 cyrschwa\or% 186  
2003 cyrnje\or% 187  
2004 cyryo\or% 188  
2005 quotedblbase\or% 189  
2006 guillemotleft\or% 190  
2007 guillemotright\or% 191  
2008 CYRA\or% 192  
2009 CYRB\or% 193  
2010 CYRV\or% 194  
2011 CYRG\or% 195  
2012 CYRD\or% 196  
2013 CYRE\or% 197  
2014 CYRZH\or% 198  
2015 CYRZ\or% 199  
2016 CYRI\or% 200

2017 CYRISHRT\or% 201  
2018 CYRK\or% 202  
2019 CYRL\or% 203  
2020 CYRM\or% 204  
2021 CYRN\or% 205  
2022 CYRO\or% 206  
2023 CYRP\or% 207  
2024 CYRR\or% 208  
2025 CYRS\or% 209  
2026 CYRT\or% 210  
2027 CYRU\or% 211  
2028 CYRF\or% 212  
2029 CYRH\or% 213  
2030 CYRC\or% 214  
2031 CYRCH\or% 215  
2032 CYRSH\or% 216  
2033 CYRSHCH\or% 217  
2034 CYRHRDSN\or% 218  
2035 CYRERY\or% 219  
2036 CYRSFTSN\or% 220  
2037 CYREREV\or% 221  
2038 CYRYU\or% 222  
2039 CYRYA\or% 223  
2040 cyra\or% 224  
2041 cyrb\or% 225  
2042 cyrv\or% 226  
2043 cyrg\or% 227  
2044 cyrd\or% 228  
2045 cyre\or% 229  
2046 cyrzh\or% 230  
2047 cyrz\or% 231  
2048 cyri\or% 232  
2049 cyrishrt\or% 233  
2050 cyrk\or% 234  
2051 cyrl\or% 235  
2052 cyrm\or% 236  
2053 cyrn\or% 237  
2054 cyro\or% 238  
2055 cyrp\or% 239  
2056 cyrr\or% 240  
2057 cyrs\or% 241  
2058 cyrt\or% 242  
2059 cyru\or% 243  
2060 cyrf\or% 244  
2061 cyrh\or% 245  
2062 cyrc\or% 246  
2063 cyrch\or% 247  
2064 cyrsh\or% 248  
2065 cyrshch\or% 249  
2066 cyrhrdsn\or% 250  
2067 cyrery\or% 251  
2068 cyrsftsn\or% 252  
2069 cyrerev\or% 253  
2070 cyryu\or% 254

```
2071 cyrya\fi% 255
2072 }
2073 </mtx & t2a>
```

### 6.3.5 T2A encoding

The T2B encoding. The data are taken from `t2b.etx`.

For unknown slots, a strange Postscript name is returned, but no warning is generated.

```
2074 <*mtx & t2b>
2075 \newcommand\getpsname[1]{%
2076   \ifcase#1%
2077   grave\or% 000
2078   acute\or% 001
2079   circumflex\or% 002
2080   tilde\or% 003
2081   dieresis\or% 004
2082   hungarumlaut\or% 005
2083   ring\or% 006
2084   caron\or% 007
2085   breve\or% 008
2086   macron\or% 009
2087   dotaccent\or% 010
2088   cedilla\or% 011
2089   ogonek\or% 012
2090   CYRpalochka\or% 013
2091   angleleft\or% 014
2092   angleright\or% 015
2093   quotedblleft\or% 016
2094   quotedblright\or% 017
2095   cyrflex\or% 018
2096   dblgrave\or% 019
2097   cyrbreve\or% 020
2098   rangedash\or% 021
2099   cyrdash\or% 022
2100   compwordmark\or% 023
2101   perthousandzero\or% 024
2102   dotlessi\or% 025
2103   dotlessj\or% 026
2104   ff\or% 027
2105   fi\or% 028
2106   fl\or% 029
2107   ffi\or% 030
2108   ffl\or% 031
2109   visiblespace\or% 032
2110   exclam\or% 033
2111   quotedbl\or% 034
2112   numbersign\or% 035
2113   dollar\or% 036
2114   percent\or% 037
2115   ampersand\or% 038
2116   quoteright\or% 039
2117   parenleft\or% 040
2118   parenright\or% 041
```

2119 asterisk\or% 042  
2120 plus\or% 043  
2121 comma\or% 044  
2122 hyphen\or% 045  
2123 period\or% 046  
2124 slash\or% 047  
2125 zero\or% 048  
2126 one\or% 049  
2127 two\or% 050  
2128 three\or% 051  
2129 four\or% 052  
2130 five\or% 053  
2131 six\or% 054  
2132 seven\or% 055  
2133 eight\or% 056  
2134 nine\or% 057  
2135 colon\or% 058  
2136 semicolon\or% 059  
2137 less\or% 060  
2138 equal\or% 061  
2139 greater\or% 062  
2140 question\or% 063  
2141 at\or% 064  
2142 A\or% 065  
2143 B\or% 066  
2144 C\or% 067  
2145 D\or% 068  
2146 E\or% 069  
2147 F\or% 070  
2148 G\or% 071  
2149 H\or% 072  
2150 I\or% 073  
2151 J\or% 074  
2152 K\or% 075  
2153 L\or% 076  
2154 M\or% 077  
2155 N\or% 078  
2156 O\or% 079  
2157 P\or% 080  
2158 Q\or% 081  
2159 R\or% 082  
2160 S\or% 083  
2161 T\or% 084  
2162 U\or% 085  
2163 V\or% 086  
2164 W\or% 087  
2165 X\or% 088  
2166 Y\or% 089  
2167 Z\or% 090  
2168 bracketleft\or% 091  
2169 backslash\or% 092  
2170 bracketright\or% 093  
2171 asciicircum\or% 094  
2172 underscore\or% 095

2173 quoteleft\or% 096  
2174 a\or% 097  
2175 b\or% 098  
2176 c\or% 099  
2177 d\or% 100  
2178 e\or% 101  
2179 f\or% 102  
2180 g\or% 103  
2181 h\or% 104  
2182 i\or% 105  
2183 j\or% 106  
2184 k\or% 107  
2185 l\or% 108  
2186 m\or% 109  
2187 n\or% 110  
2188 o\or% 111  
2189 p\or% 112  
2190 q\or% 113  
2191 r\or% 114  
2192 s\or% 115  
2193 t\or% 116  
2194 u\or% 117  
2195 v\or% 118  
2196 w\or% 119  
2197 x\or% 120  
2198 y\or% 121  
2199 z\or% 122  
2200 braceleft\or% 123  
2201 bar\or% 124  
2202 braceright\or% 125  
2203 asciitilde\or% 126  
2204 hyphenchar\or% 127  
2205 CYRGDSCHCRS\or% 128  
2206 CYRGHCRS\or% 129  
2207 CYRGDSC\or% 130  
2208 CYRGHK\or% 131  
2209 CYRSHHA\or% 132  
2210 CYRZHDSC\or% 133  
2211 CYRDELTA\or% 134  
2212 CYRABHDZE\or% 135  
2213 CYRLJE\or% 136  
2214 CYRKDSC\or% 137  
2215 CYRLDSC\or% 138  
2216 CYRKHK\or% 139  
2217 CYRLHK\or% 140  
2218 CYRNDSC\or% 141  
2219 CYRNG\or% 142  
2220 CYRNHK\or% 143  
2221 CYROTLD\or% 144  
2222 CYRSACRS\or% 145  
2223 CYRUSHRT\or% 146  
2224 CYRY\or% 147  
2225 CYRHHCRS\or% 148  
2226 CYRHDSC\or% 149

2227 CYRHHK\or% 150  
2228 CYRCHLDSC\or% 151  
2229 CYRCHRDSC\or% 152  
2230 CYRNJE\or% 153  
2231 CYRSCHWA\or% 154  
2232 CYREPS\or% 155  
2233 CYRYO\or% 156  
2234 numero\or% 157  
2235 currency\or% 158  
2236 section\or% 159  
2237 cyrgdschcrs\or% 160  
2238 cyrghcrs\or% 161  
2239 cyrgdsc\or% 162  
2240 cyrghk\or% 163  
2241 cyrshha\or% 164  
2242 cyrzhdsc\or% 165  
2243 cyrdelta\or% 166  
2244 cyrabhdze\or% 167  
2245 cyrlje\or% 168  
2246 cyrkdsc\or% 169  
2247 cyrldsc\or% 170  
2248 cyrkhk\or% 171  
2249 cyrlhk\or% 172  
2250 cyrndsc\or% 173  
2251 cyrng\or% 174  
2252 cyrnhk\or% 175  
2253 cyrotld\or% 176  
2254 cyrsacrs\or% 177  
2255 cyrushrt\or% 178  
2256 cyry\or% 179  
2257 cyrhhcrs\or% 180  
2258 cyrhdsc\or% 181  
2259 cyrhk\or% 182  
2260 cyrchldsc\or% 183  
2261 cyrchrsc\or% 184  
2262 cyrnje\or% 185  
2263 cyrschwa\or% 186  
2264 cyreps\or% 187  
2265 cyryo\or% 188  
2266 quotedblbase\or% 189  
2267 guillemotleft\or% 190  
2268 guillemotright\or% 191  
2269 CYRA\or% 192  
2270 CYRB\or% 193  
2271 CYRV\or% 194  
2272 CYRG\or% 195  
2273 CYRD\or% 196  
2274 CYRE\or% 197  
2275 CYRZH\or% 198  
2276 CYRZ\or% 199  
2277 CYRI\or% 200  
2278 CYRISHRT\or% 201  
2279 CYRK\or% 202  
2280 CYRL\or% 203

2281 CYRM\or% 204  
2282 CYRN\or% 205  
2283 CYRO\or% 206  
2284 CYRP\or% 207  
2285 CYRR\or% 208  
2286 CYRS\or% 209  
2287 CYRT\or% 210  
2288 CYRU\or% 211  
2289 CYRF\or% 212  
2290 CYRH\or% 213  
2291 CYRC\or% 214  
2292 CYRCH\or% 215  
2293 CYRSH\or% 216  
2294 CYRSHCH\or% 217  
2295 CYRHRDSN\or% 218  
2296 CYRERY\or% 219  
2297 CYRSFTSN\or% 220  
2298 CYREREV\or% 221  
2299 CYRYU\or% 222  
2300 CYRYA\or% 223  
2301 cyra\or% 224  
2302 cyrb\or% 225  
2303 cyrv\or% 226  
2304 cyrg\or% 227  
2305 cyrd\or% 228  
2306 cyre\or% 229  
2307 cyrzh\or% 230  
2308 cyrz\or% 231  
2309 cyri\or% 232  
2310 cyrishrt\or% 233  
2311 cyrk\or% 234  
2312 cyrl\or% 235  
2313 cyrm\or% 236  
2314 cyrn\or% 237  
2315 cyro\or% 238  
2316 cyrp\or% 239  
2317 cyrr\or% 240  
2318 cyrs\or% 241  
2319 cyrt\or% 242  
2320 cyru\or% 243  
2321 cyrf\or% 244  
2322 cyrh\or% 245  
2323 cyrc\or% 246  
2324 cyrch\or% 247  
2325 cyrsh\or% 248  
2326 cyrshch\or% 249  
2327 cyrhrdsn\or% 250  
2328 cyrery\or% 251  
2329 cyrsftsn\or% 252  
2330 cyrerev\or% 253  
2331 cyryu\or% 254  
2332 cyrya\fi% 255  
2333 }  
2334 </mtx & t2b)

### 6.3.6 LY1 encoding

The LY1 encoding. The data are taken from `texnansi.enc`.

For unknown slots, a strange Postscript name is returned, but no warning is generated.

```
2335 (*mtx & ly1)
2336 \newcommand\getpsname[1]{%
2337   \ifcase#1%
2338   .notdef.000\or % 000
2339   Euro\or % 001
2340   .notdef.002\or % 002
2341   .notdef.003\or % 003
2342   fraction\or % 004
2343   dotaccent\or % 005
2344   hungarumlaut\or % 006
2345   ogonek\or % 007
2346   fl\or % 008
2347   .notdef.009\or % 009
2348   cwm\or % 010
2349   ff\or % 011
2350   fi\or % 012
2351   .notdef.013\or % 013
2352   ffi\or % 014
2353   ffl\or % 015
2354   dotlessi\or % 016
2355   dotlessj\or % 017
2356   grave\or % 018
2357   acute\or % 019
2358   caron\or % 020
2359   breve\or % 021
2360   macron\or % 022
2361   ring\or % 023
2362   cedilla\or % 024
2363   germandbls\or % 025
2364   ae\or % 026
2365   oe\or % 027
2366   oslash\or % 028
2367   AE\or % 029
2368   OE\or % 030
2369   Oslash\or % 031
2370   space\or % 032
2371   exclam\or % 033
2372   quotedbl\or % 034
2373   numbersign\or % 035
2374   dollar\or % 036
2375   percent\or % 037
2376   ampersand\or % 038
2377   quoteright\or % 039
2378   parenleft\or % 040
2379   parenright\or % 041
2380   asterisk\or % 042
2381   plus\or % 043
2382   comma\or % 044
2383   hyphen\or % 045
```

2384 period\or % 046  
2385 slash\or % 047  
2386 zero\or % 048  
2387 one\or % 049  
2388 two\or % 050  
2389 three\or % 051  
2390 four\or % 052  
2391 five\or % 053  
2392 six\or % 054  
2393 seven\or % 055  
2394 eight\or % 056  
2395 nine\or % 057  
2396 colon\or % 058  
2397 semicolon\or % 059  
2398 less\or % 060  
2399 equal\or % 061  
2400 greater\or % 062  
2401 question\or % 063  
2402 at\or % 064  
2403 A\or % 065  
2404 B\or % 066  
2405 C\or % 067  
2406 D\or % 068  
2407 E\or % 069  
2408 F\or % 070  
2409 G\or % 071  
2410 H\or % 072  
2411 I\or % 073  
2412 J\or % 074  
2413 K\or % 075  
2414 L\or % 076  
2415 M\or % 077  
2416 N\or % 078  
2417 O\or % 079  
2418 P\or % 080  
2419 Q\or % 081  
2420 R\or % 082  
2421 S\or % 083  
2422 T\or % 084  
2423 U\or % 085  
2424 V\or % 086  
2425 W\or % 087  
2426 X\or % 088  
2427 Y\or % 089  
2428 Z\or % 090  
2429 bracketleft\or % 091  
2430 backslash\or % 092  
2431 bracketright\or % 093  
2432 circumflex\or % 094  
2433 underscore\or % 095  
2434 quoteleft\or % 096  
2435 a\or % 097  
2436 b\or % 098  
2437 c\or % 099

2438 d\or % 100  
 2439 e\or % 101  
 2440 f\or % 102  
 2441 g\or % 103  
 2442 h\or % 104  
 2443 i\or % 105  
 2444 j\or % 106  
 2445 k\or % 107  
 2446 l\or % 108  
 2447 m\or % 109  
 2448 n\or % 110  
 2449 o\or % 111  
 2450 p\or % 112  
 2451 q\or % 113  
 2452 r\or % 114  
 2453 s\or % 115  
 2454 t\or % 116  
 2455 u\or % 117  
 2456 v\or % 118  
 2457 w\or % 119  
 2458 x\or % 120  
 2459 y\or % 121  
 2460 z\or % 122  
 2461 braceleft\or % 123  
 2462 bar\or % 124  
 2463 braceright\or % 125  
 2464 tilde\or % 126  
 2465 dieresis\or % 127  
 2466 Lslash\or % 128  
 2467 quotesingle\or % 129  
 2468 quotesinglbase\or % 130  
 2469 florin\or % 131  
 2470 quotedblbase\or % 132  
 2471 ellipsis\or % 133  
 2472 dagger\or % 134  
 2473 daggerdbl\or % 135  
 2474 circumflex\or % 136  
 2475 perthousand\or % 137  
 2476 Scaron\or % 138  
 2477 guilsinglleft\or % 139  
 2478 OE\or % 140  
 2479 Zcaron\or % 141  
 2480 asciicircum\or % 142  
 2481 minus\or % 143  
 2482 lslash\or % 144  
 2483 quoteleft\or % 145  
 2484 quoteright\or % 146  
 2485 quotedblleft\or % 147  
 2486 quotedblright\or % 148  
 2487 bullet\or % 149  
 2488 endash\or % 150  
 2489 emdash\or % 151  
 2490 tilde\or % 152  
 2491 trademark\or % 153

2492 scaron\or % 154  
2493 guilsinglright\or % 155  
2494 oe\or % 156  
2495 zcaron\or % 157  
2496 asciitilde\or % 158  
2497 Ydieresis\or % 159  
2498 nbspace\or % 160  
2499 exclamdown\or % 161  
2500 cent\or % 162  
2501 sterling\or % 163  
2502 currency\or % 164  
2503 yen\or % 165  
2504 brokenbar\or % 166  
2505 section\or % 167  
2506 dieresis\or % 168  
2507 copyright\or % 169  
2508 ordfeminine\or % 170  
2509 guillemotleft\or % 171  
2510 logicalnot\or % 172  
2511 sfthyphen\or % 173  
2512 registered\or % 174  
2513 macron\or % 175  
2514 degree\or % 176  
2515 plusminus\or % 177  
2516 twosuperior\or % 178  
2517 threesuperior\or % 179  
2518 acute\or % 180  
2519 mu\or % 181  
2520 paragraph\or % 182  
2521 periodcentered\or % 183  
2522 cedilla\or % 184  
2523 onesuperior\or % 185  
2524 ordmasculine\or % 186  
2525 guillemotright\or % 187  
2526 onequarter\or % 188  
2527 onehalf\or % 189  
2528 threequarters\or % 190  
2529 questiondown\or % 191  
2530 Agrave\or % 192  
2531 Aacute\or % 193  
2532 Acircumflex\or % 194  
2533 Atilde\or % 195  
2534 Adieresis\or % 196  
2535 Aring\or % 197  
2536 AE\or % 198  
2537 Ccedilla\or % 199  
2538 Egrave\or % 200  
2539 Eacute\or % 201  
2540 Ecircumflex\or % 202  
2541 Edieresis\or % 203  
2542 Igrave\or % 204  
2543 Iacute\or % 205  
2544 Icircumflex\or % 206  
2545 Idieresis\or % 207

```

2546 Eth\or % 208
2547 Ntilde\or % 209
2548 Ograve\or % 210
2549 Oacute\or % 211
2550 Ocircumflex\or % 212
2551 Otilde\or % 213
2552 Odieresis\or % 214
2553 multiply\or % 215
2554 Oslash\or % 216
2555 Ugrave\or % 217
2556 Uacute\or % 218
2557 Ucircumflex\or % 219
2558 Udieresis\or % 220
2559 Yacute\or % 221
2560 Thorn\or % 222
2561 germandbls\or % 223
2562 agrave\or % 224
2563 aacute\or % 225
2564 acircumflex\or % 226
2565 atilde\or % 227
2566 adieresis\or % 228
2567 aring\or % 229
2568 ae\or % 230
2569 ccedilla\or % 231
2570 egrave\or % 232
2571 eacute\or % 233
2572 ecircumflex\or % 234
2573 edieresis\or % 235
2574 igrave\or % 236
2575 iacute\or % 237
2576 icircumflex\or % 238
2577 idieresis\or % 239
2578 eth\or % 240
2579 ntilde\or % 241
2580 ograve\or % 242
2581 oacute\or % 243
2582 ocircumflex\or % 244
2583 otilde\or % 245
2584 odieresis\or % 246
2585 divide\or % 247
2586 oslash\or % 248
2587 ugrave\or % 249
2588 uacute\or % 250
2589 ucircumflex\or % 251
2590 udieresis\or % 252
2591 yacute\or % 253
2592 thorn\or % 254
2593 ydieresis\fi % 255
2594 }
2595 </mtx & ly1>

```

## 6.4 Templates

Generate template files for T1 and TS1 encoding. They should be self-describing.

## 6.4.1 T1 encoding

```
2596 (*template & t1)
2597 \listfiles
2598 %% Replace the 'XXX' in the next line by the 3- or 4-character long
2599 %% abbreviation for your font.
2600 \documentclass[family=XXX]{kerntest}
2601
2602 %% Replace the settings by these you want to test.
2603 \kernsetup{encoding=T1,series=m,shape=n,example=hello}
2604 \kernsetup{size=14.40pt,baselineskip=16.5pt,papersize=a4paper}
2605
2606 %% The next line can be used to add a name suffix to the output |mtx| file.
2607 %% \kernsetup{extraname=normal}
2608
2609 %% Set encoding parameters.
2610 %% Set ligaturing: 1=all, 0=all, -1=some, -2=none
2611 %% \encodingsetup{ligaturing=1}
2612
2613 %% If you are using a font with different design sizes and if you want
2614 %% to test one design size scaled to another one, you may input a
2615 %% modified fd file for your font. To generate this new fd file, just
2616 %% copy the original one, rename it, and modify the entries for the
2617 %% font shapes to use the design size you want to test.
2618 %% \input{t1XXX-1200.fd}
2619
2620 %% The following lines show some possible glyph classes. You should
2621 %% add all classes you need.
2622 \newglyphclass{right}{A}{A,Aogonek}
2623 \newglyphclass{right}{Abreve}{Abreve,Aacute,Acircumflex,%
2624   Atilde,Adieresis}
2625 \newglyphclass{right}{Aring}{Agrave,Aring}
2626
2627 \newglyphclass{left}{A}{A,Aogonek}
2628 \newglyphclass{left}{Abreve}{Abreve,Agrave,Acircumflex,%
2629   Atilde,Adieresis}
2630 \newglyphclass{left}{Aring}{Aacute,Aring}
2631
2632 \newglyphclass{right}{C}{C,Cacute,Ccaron,Ccedilla}
2633 \newglyphclass{left}{C}{C,Cacute,Ccaron,Ccedilla}
2634
2635 \newglyphclass{right}{D}{D,Dcaron,Eth}
2636
2637 \newglyphclass{right}{E}{E,Ecaron,Eogonek,Egrave,Eacute,Ecircumflex,%
2638   Edieresis,AE,OE}
2639
2640 \newglyphclass{right}{G}{G,Gbreve}
2641 \newglyphclass{left}{G}{G,Gbreve}
2642
2643 \newglyphclass{left}{H}{B,D,Dcaron,Eth,E,Ecaron,Eogonek,Egrave,%
2644   Eacute,Ecircumflex,Edieresis,F,H,I,Idotaccent,Igrave,Iacute,%
2645   Icircumflex,Idieresis,IJ,J,K,L,Lacute,Lcaron,N,Nacute,Ncaron,%
2646   Ntilde,P,R,Racute,Rcaron,Ng,Thorn}
2647
```

2648 \newglyphclass{right}{H}{H,I,Idotaccent,Igrave,Iacute,%  
2649 Icircumflex,Idieresis,IJ,J,N,Nacute,Ncaron,Ntilde}  
2650  
2651 \newglyphclass{right}{O}{O,Ohungarumlaut,Ograve,Oacute,Ocircumflex,%  
2652 Otilde,Odieresis,Oslash}  
2653 \newglyphclass{left}{O}{O,Ohungarumlaut,Ograve,Oacute,Ocircumflex,%  
2654 Otilde,Odieresis,OE,Oslash}  
2655  
2656 \newglyphclass{right}{R}{R,Racute,Rcaron}  
2657  
2658 \newglyphclass{right}{S}{S,Sacute,Scaron,Scedilla,SS,dollar}  
2659 \newglyphclass{left}{S}{S,Sacute,Scaron,Scedilla,SS,dollar}  
2660  
2661 \newglyphclass{right}{T}{T,Tcaron,Tcedilla}  
2662 \newglyphclass{left}{T}{T,Tcaron,Tcedilla}  
2663  
2664 \newglyphclass{right}{U}{U,Uhungarumlaut,Uring,Ugrave,Uacute,%  
2665 Ucircumflex,Udieresis}  
2666 \newglyphclass{left}{U}{U,Uhungarumlaut,Uring,Ugrave,Uacute,%  
2667 Ucircumflex,Udieresis}  
2668  
2669 \newglyphclass{right}{Y}{Y,Yacute,Ydieresis}%  
2670 \newglyphclass{left}{Y}{Y,Yacute,Ydieresis}%  
2671  
2672 \newglyphclass{left}{Z}{Z,Zacute,Zcaron,Zdotaccent}  
2673 \newglyphclass{right}{Z}{Z,Zacute,Zcaron,Zdotaccent}  
2674  
2675 \newglyphclass{left}{a}{a,aogonek,ae}  
2676 \newglyphclass{left}{abreve}{abreve,agrave,acircumflex,atilde,adieresis,aring}  
2677 \newglyphclass{right}{a}{a,aogonek}  
2678 \newglyphclass{right}{abreve}{abreve,aacute,acircumflex,atilde,adieresis,aring}  
2679  
2680 \newglyphclass{left}{c}{c,ccedilla}  
2681 \newglyphclass{right}{c}{c,ccedilla}  
2682  
2683 \newglyphclass{left}{d}{d,dbar,dcaron}  
2684  
2685 \newglyphclass{left}{e}{e,eogonek}  
2686 \newglyphclass{left}{egrave}{ecaron,ecircumflex}  
2687 \newglyphclass{right}{e}{e,eogonek,ae,oe}  
2688 \newglyphclass{right}{egrave}{ecaron,ecircumflex}  
2689  
2690 \newglyphclass{left}{f}{f,ff,fi,fl,ffi,ffl}  
2691 \newglyphclass{right}{f}{f,ff}  
2692  
2693 \newglyphclass{right}{i}{i,fi,ffi,dotlessi}  
2694 \newglyphclass{left}{i}{i,ij,dotlessi}  
2695  
2696 \newglyphclass{right}{l}{fl,ffl,l,lacute}  
2697 \newglyphclass{left}{l}{l,lacute,lcaron}  
2698  
2699 \newglyphclass{right}{m}{m,n}  
2700 \newglyphclass{left}{m}{m,n}  
2701 \newglyphclass{right}{nacute}{nacute,ncaron}

```

2702 \newglyphclass{left}{nacute}{nacute,ncaron}
2703
2704 \newglyphclass{right}{ograve}{ograve,ocircumflex}
2705 \newglyphclass{right}{ohungarumlaut}{ohungarumlaut,oacute,otilde,%
2706 odieresis}
2707 \newglyphclass{left}{oacute}{oacute,ocircumflex,oe}
2708 \newglyphclass{left}{ohungarumlaut}{ohungarumlaut,ograve,otilde,%
2709 odieresis}
2710
2711 \newglyphclass{left}{t}{t,tcaron,tcedilla}
2712
2713
2714 \begin{document}
2715
2716 %% This table of characters is sorted by similar glyphs, not by the
2717 %% encoding.
2718 %% Replace ‘{LLL}’ and ‘{RRR}’ in columns 1 resp. 5 by these glyphs
2719 %% that shall be tested.
2720 %% Good pairs to use for LLL and RRR test are 016, 017 (English
2721 %% quotation marks ‘ and ’); 018, 016 (German quotation marks , , and
2722 %% ‘); 019, 020 (French Guillemets << and >>); 020, 019 (French
2723 %% Guillemets in German notation >> and <<).
2724 %% You may also take the Postscript names for all glyphs.
2725 \begin{kerntable}
2726 \testkern{LLL}{-}{000}{-}{RRR} \ \ % grave
2727 \testkern{LLL}{-}{001}{-}{RRR} \ \ % acute
2728 \testkern{LLL}{-}{002}{-}{RRR} \ \ % circumflex
2729 \testkern{LLL}{-}{003}{-}{RRR} \ \ % tilde
2730 \testkern{LLL}{-}{004}{-}{RRR} \ \ % dieresis
2731 \testkern{LLL}{-}{005}{-}{RRR} \ \ % hungarumlaut
2732 \testkern{LLL}{-}{006}{-}{RRR} \ \ % ring
2733 \testkern{LLL}{-}{007}{-}{RRR} \ \ % caron
2734 \testkern{LLL}{-}{008}{-}{RRR} \ \ % breve
2735 \testkern{LLL}{-}{009}{-}{RRR} \ \ % macron
2736 \testkern{LLL}{-}{010}{-}{RRR} \ \ % dotaccent
2737 \testkern{LLL}{-}{011}{-}{RRR} \ \ % cedilla
2738 \testkern{LLL}{-}{012}{-}{RRR} \ \ % ogonek
2739 \testkern{LLL}{-}{126}{-}{RRR} \ \ % asciitilde
2740 \testkern{LLL}{-}{094}{-}{RRR} \ \ % asciiicircum
2741 \testkern{LLL}{-}{042}{-}{RRR} \ \ % asterisk
2742 \testkern{LLL}{-}{023}{-}{RRR} \ \ % compwordmark
2743 \testkern{LLL}{-}{032}{-}{RRR} \ \ % visiblespace
2744 \testkern{LLL}{-}{014}{-}{RRR} \ \ % guilsinglleft
2745 \testkern{LLL}{-}{015}{-}{RRR} \ \ % guilsinglright
2746 \testkern{LLL}{-}{019}{-}{RRR} \ \ % guillemotleft
2747 \testkern{LLL}{-}{020}{-}{RRR} \ \ % guillemotright
2748 \testkern{LLL}{-}{096}{-}{RRR} \ \ % quoteleft
2749 \testkern{LLL}{-}{039}{-}{RRR} \ \ % quoteright
2750 \testkern{LLL}{-}{013}{-}{RRR} \ \ % quotesinglbase
2751 \testkern{LLL}{-}{016}{-}{RRR} \ \ % quotedblleft
2752 \testkern{LLL}{-}{017}{-}{RRR} \ \ % quotedblright
2753 \testkern{LLL}{-}{018}{-}{RRR} \ \ % quotedblbase
2754 \testkern{LLL}{-}{034}{-}{RRR} \ \ % quotedbl
2755 \testkern{LLL}{-}{046}{-}{RRR} \ \ % period

```

2756 \testkern{LLL}{-}{044}{-}{RRR} \\ % comma  
2757 \testkern{LLL}{-}{058}{-}{RRR} \\ % colon  
2758 \testkern{LLL}{-}{059}{-}{RRR} \\ % semicolon  
2759 \testkern{LLL}{-}{033}{-}{RRR} \\ % exclam  
2760 \testkern{LLL}{-}{063}{-}{RRR} \\ % question  
2761 \testkern{LLL}{-}{189}{-}{RRR} \\ % exclamdown  
2762 \testkern{LLL}{-}{190}{-}{RRR} \\ % questiondown  
2763 \testkern{LLL}{-}{045}{-}{RRR} \\ % hyphen  
2764 \testkern{LLL}{-}{127}{-}{RRR} \\ % hyphenchar  
2765 \testkern{LLL}{-}{021}{-}{RRR} \\ % rangedash  
2766 \testkern{LLL}{-}{022}{-}{RRR} \\ % punctdash  
2767 \testkern{LLL}{-}{095}{-}{RRR} \\ % underscore  
2768 \testkern{LLL}{-}{043}{-}{RRR} \\ % plus  
2769 \testkern{LLL}{-}{061}{-}{RRR} \\ % equal  
2770 \testkern{LLL}{-}{060}{-}{RRR} \\ % less  
2771 \testkern{LLL}{-}{062}{-}{RRR} \\ % greater  
2772 \testkern{LLL}{-}{047}{-}{RRR} \\ % slash  
2773 \testkern{LLL}{-}{092}{-}{RRR} \\ % backslash  
2774 \testkern{LLL}{-}{040}{-}{RRR} \\ % parenleft  
2775 \testkern{LLL}{-}{041}{-}{RRR} \\ % parenright  
2776 \testkern{LLL}{-}{091}{-}{RRR} \\ % bracketleft  
2777 \testkern{LLL}{-}{093}{-}{RRR} \\ % bracketright  
2778 \testkern{LLL}{-}{123}{-}{RRR} \\ % braceleft  
2779 \testkern{LLL}{-}{125}{-}{RRR} \\ % braceright  
2780 \testkern{LLL}{-}{124}{-}{RRR} \\ % bar  
2781 \testkern{LLL}{-}{035}{-}{RRR} \\ % numbersign  
2782 \testkern{LLL}{-}{037}{-}{RRR} \\ % percent  
2783 \testkern{LLL}{-}{024}{-}{RRR} \\ % perthousandzero  
2784 \testkern{LLL}{-}{038}{-}{RRR} \\ % ampersand  
2785 \testkern{LLL}{-}{159}{-}{RRR} \\ % section  
2786 \testkern{LLL}{-}{064}{-}{RRR} \\ % at  
2787 \testkern{LLL}{-}{191}{-}{RRR} \\ % sterling  
2788 \testkern{LLL}{-}{048}{-}{RRR} \\ % zero  
2789 \testkern{LLL}{-}{049}{-}{RRR} \\ % one  
2790 \testkern{LLL}{-}{050}{-}{RRR} \\ % two  
2791 \testkern{LLL}{-}{051}{-}{RRR} \\ % three  
2792 \testkern{LLL}{-}{052}{-}{RRR} \\ % four  
2793 \testkern{LLL}{-}{053}{-}{RRR} \\ % five  
2794 \testkern{LLL}{-}{054}{-}{RRR} \\ % six  
2795 \testkern{LLL}{-}{055}{-}{RRR} \\ % seven  
2796 \testkern{LLL}{-}{056}{-}{RRR} \\ % eight  
2797 \testkern{LLL}{-}{057}{-}{RRR} \\ % nine  
2798 \testkern{LLL}{-}{065}{-}{RRR} \\ % A  
2799 \testkern{LLL}{-}{128}{-}{RRR} \\ % Abreve  
2800 \testkern{LLL}{-}{129}{-}{RRR} \\ % Aogonek  
2801 \testkern{LLL}{-}{192}{-}{RRR} \\ % Agrave  
2802 \testkern{LLL}{-}{193}{-}{RRR} \\ % Acute  
2803 \testkern{LLL}{-}{194}{-}{RRR} \\ % Acircumflex  
2804 \testkern{LLL}{-}{195}{-}{RRR} \\ % Atilde  
2805 \testkern{LLL}{-}{196}{-}{RRR} \\ % Adieresis  
2806 \testkern{LLL}{-}{197}{-}{RRR} \\ % Aring  
2807 \testkern{LLL}{-}{198}{-}{RRR} \\ % AE  
2808 \testkern{LLL}{-}{066}{-}{RRR} \\ % B  
2809 \testkern{LLL}{-}{067}{-}{RRR} \\ % C

2810 \testkern{LLL}{-}{130}{-}{RRR} \\ % Cacute  
2811 \testkern{LLL}{-}{131}{-}{RRR} \\ % Ccaron  
2812 \testkern{LLL}{-}{199}{-}{RRR} \\ % Ccedilla  
2813 \testkern{LLL}{-}{068}{-}{RRR} \\ % D  
2814 \testkern{LLL}{-}{132}{-}{RRR} \\ % Dcaron  
2815 \testkern{LLL}{-}{208}{-}{RRR} \\ % Eth  
2816 \testkern{LLL}{-}{069}{-}{RRR} \\ % E  
2817 \testkern{LLL}{-}{133}{-}{RRR} \\ % Ecaron  
2818 \testkern{LLL}{-}{134}{-}{RRR} \\ % Eogonek  
2819 \testkern{LLL}{-}{200}{-}{RRR} \\ % Egrave  
2820 \testkern{LLL}{-}{201}{-}{RRR} \\ % Eacute  
2821 \testkern{LLL}{-}{202}{-}{RRR} \\ % Ecircumflex  
2822 \testkern{LLL}{-}{203}{-}{RRR} \\ % Edieresis  
2823 \testkern{LLL}{-}{070}{-}{RRR} \\ % F  
2824 \testkern{LLL}{-}{071}{-}{RRR} \\ % G  
2825 \testkern{LLL}{-}{135}{-}{RRR} \\ % Gbreve  
2826 \testkern{LLL}{-}{072}{-}{RRR} \\ % H  
2827 \testkern{LLL}{-}{073}{-}{RRR} \\ % I  
2828 \testkern{LLL}{-}{157}{-}{RRR} \\ % Idotaccent  
2829 \testkern{LLL}{-}{204}{-}{RRR} \\ % Igrave  
2830 \testkern{LLL}{-}{205}{-}{RRR} \\ % Iacute  
2831 \testkern{LLL}{-}{206}{-}{RRR} \\ % Icircumflex  
2832 \testkern{LLL}{-}{207}{-}{RRR} \\ % Idieresis  
2833 \testkern{LLL}{-}{156}{-}{RRR} \\ % IJ  
2834 \testkern{LLL}{-}{074}{-}{RRR} \\ % J  
2835 \testkern{LLL}{-}{075}{-}{RRR} \\ % K  
2836 \testkern{LLL}{-}{076}{-}{RRR} \\ % L  
2837 \testkern{LLL}{-}{136}{-}{RRR} \\ % Lacute  
2838 \testkern{LLL}{-}{137}{-}{RRR} \\ % Lcaron  
2839 \testkern{LLL}{-}{138}{-}{RRR} \\ % Lslash  
2840 \testkern{LLL}{-}{077}{-}{RRR} \\ % M  
2841 \testkern{LLL}{-}{078}{-}{RRR} \\ % N  
2842 \testkern{LLL}{-}{139}{-}{RRR} \\ % Nacute  
2843 \testkern{LLL}{-}{140}{-}{RRR} \\ % Ncaron  
2844 \testkern{LLL}{-}{209}{-}{RRR} \\ % Ntilde  
2845 \testkern{LLL}{-}{079}{-}{RRR} \\ % O  
2846 \testkern{LLL}{-}{142}{-}{RRR} \\ % Ohungarumlaut  
2847 \testkern{LLL}{-}{210}{-}{RRR} \\ % Ograve  
2848 \testkern{LLL}{-}{211}{-}{RRR} \\ % Oacute  
2849 \testkern{LLL}{-}{212}{-}{RRR} \\ % Ocircumflex  
2850 \testkern{LLL}{-}{213}{-}{RRR} \\ % Otilde  
2851 \testkern{LLL}{-}{214}{-}{RRR} \\ % Odieresis  
2852 \testkern{LLL}{-}{215}{-}{RRR} \\ % OE  
2853 \testkern{LLL}{-}{216}{-}{RRR} \\ % Oslash  
2854 \testkern{LLL}{-}{080}{-}{RRR} \\ % P  
2855 \testkern{LLL}{-}{081}{-}{RRR} \\ % Q  
2856 \testkern{LLL}{-}{082}{-}{RRR} \\ % R  
2857 \testkern{LLL}{-}{143}{-}{RRR} \\ % Racute  
2858 \testkern{LLL}{-}{144}{-}{RRR} \\ % Rcaron  
2859 \testkern{LLL}{-}{083}{-}{RRR} \\ % S  
2860 \testkern{LLL}{-}{145}{-}{RRR} \\ % Sacute  
2861 \testkern{LLL}{-}{146}{-}{RRR} \\ % Scaron  
2862 \testkern{LLL}{-}{147}{-}{RRR} \\ % Scedilla  
2863 \testkern{LLL}{-}{223}{-}{RRR} \\ % SS

2864 \testkern{LLL}{-}{036}{-}{RRR} \\ % dollar  
2865 \testkern{LLL}{-}{084}{-}{RRR} \\ % T  
2866 \testkern{LLL}{-}{148}{-}{RRR} \\ % Tcaron  
2867 \testkern{LLL}{-}{149}{-}{RRR} \\ % Tcedilla  
2868 \testkern{LLL}{-}{085}{-}{RRR} \\ % U  
2869 \testkern{LLL}{-}{150}{-}{RRR} \\ % Ungarumlaut  
2870 \testkern{LLL}{-}{151}{-}{RRR} \\ % Uring  
2871 \testkern{LLL}{-}{217}{-}{RRR} \\ % Ugrave  
2872 \testkern{LLL}{-}{218}{-}{RRR} \\ % Uacute  
2873 \testkern{LLL}{-}{219}{-}{RRR} \\ % Ucircumflex  
2874 \testkern{LLL}{-}{220}{-}{RRR} \\ % Udieresis  
2875 \testkern{LLL}{-}{086}{-}{RRR} \\ % V  
2876 \testkern{LLL}{-}{087}{-}{RRR} \\ % W  
2877 \testkern{LLL}{-}{088}{-}{RRR} \\ % X  
2878 \testkern{LLL}{-}{089}{-}{RRR} \\ % Y  
2879 \testkern{LLL}{-}{152}{-}{RRR} \\ % Ydieresis  
2880 \testkern{LLL}{-}{221}{-}{RRR} \\ % Yacute  
2881 \testkern{LLL}{-}{090}{-}{RRR} \\ % Z  
2882 \testkern{LLL}{-}{153}{-}{RRR} \\ % Zacute  
2883 \testkern{LLL}{-}{154}{-}{RRR} \\ % Zcaron  
2884 \testkern{LLL}{-}{155}{-}{RRR} \\ % Zdotaccent  
2885 \testkern{LLL}{-}{141}{-}{RRR} \\ % Ng  
2886 \testkern{LLL}{-}{222}{-}{RRR} \\ % Thorn  
2887 \testkern{LLL}{-}{097}{-}{RRR} \\ % a  
2888 \testkern{LLL}{-}{160}{-}{RRR} \\ % abreve  
2889 \testkern{LLL}{-}{161}{-}{RRR} \\ % aogonek  
2890 \testkern{LLL}{-}{224}{-}{RRR} \\ % agrave  
2891 \testkern{LLL}{-}{225}{-}{RRR} \\ % aacute  
2892 \testkern{LLL}{-}{226}{-}{RRR} \\ % acircumflex  
2893 \testkern{LLL}{-}{227}{-}{RRR} \\ % atilde  
2894 \testkern{LL}{-}{228}{-}{RRR} \\ % adieresis  
2895 \testkern{LLL}{-}{229}{-}{RRR} \\ % aring  
2896 \testkern{LLL}{-}{230}{-}{RRR} \\ % ae  
2897 \testkern{LLL}{-}{098}{-}{RRR} \\ % b  
2898 \testkern{LLL}{-}{099}{-}{RRR} \\ % c  
2899 \testkern{LLL}{-}{162}{-}{RRR} \\ % cacute  
2900 \testkern{LLL}{-}{163}{-}{RRR} \\ % ccaron  
2901 \testkern{LLL}{-}{231}{-}{RRR} \\ % ccedilla  
2902 \testkern{LLL}{-}{100}{-}{RRR} \\ % d  
2903 \testkern{LLL}{-}{158}{-}{RRR} \\ % dbar  
2904 \testkern{LLL}{-}{164}{-}{RRR} \\ % dcaron  
2905 \testkern{LLL}{-}{101}{-}{RRR} \\ % e  
2906 \testkern{LLL}{-}{165}{-}{RRR} \\ % ecaron  
2907 \testkern{LLL}{-}{166}{-}{RRR} \\ % eogonek  
2908 \testkern{LLL}{-}{232}{-}{RRR} \\ % egrave  
2909 \testkern{LLL}{-}{233}{-}{RRR} \\ % eacute  
2910 \testkern{LLL}{-}{234}{-}{RRR} \\ % ecircumflex  
2911 \testkern{LLL}{-}{235}{-}{RRR} \\ % edieresis  
2912 \testkern{LLL}{-}{102}{-}{RRR} \\ % f  
2913 \testkern{LLL}{-}{027}{-}{RRR} \\ % ff  
2914 \testkern{LLL}{-}{028}{-}{RRR} \\ % fi  
2915 \testkern{LLL}{-}{029}{-}{RRR} \\ % fl  
2916 \testkern{LLL}{-}{030}{-}{RRR} \\ % ffi  
2917 \testkern{LLL}{-}{031}{-}{RRR} \\ % ffl

2918 \testkern{LLL}{-}{103}{-}{RRR} \\ % g  
2919 \testkern{LLL}{-}{167}{-}{RRR} \\ % gbreve  
2920 \testkern{LLL}{-}{104}{-}{RRR} \\ % h  
2921 \testkern{LLL}{-}{105}{-}{RRR} \\ % i  
2922 \testkern{LLL}{-}{025}{-}{RRR} \\ % dotlessi  
2923 \testkern{LLL}{-}{236}{-}{RRR} \\ % igrave  
2924 \testkern{LLL}{-}{237}{-}{RRR} \\ % iacute  
2925 \testkern{LLL}{-}{238}{-}{RRR} \\ % icircumflex  
2926 \testkern{LLL}{-}{239}{-}{RRR} \\ % idieresis  
2927 \testkern{LLL}{-}{188}{-}{RRR} \\ % ij  
2928 \testkern{LLL}{-}{106}{-}{RRR} \\ % j  
2929 \testkern{LLL}{-}{026}{-}{RRR} \\ % dotlessj  
2930 \testkern{LLL}{-}{107}{-}{RRR} \\ % k  
2931 \testkern{LLL}{-}{108}{-}{RRR} \\ % l  
2932 \testkern{LLL}{-}{168}{-}{RRR} \\ % lacute  
2933 \testkern{LLL}{-}{169}{-}{RRR} \\ % lcaron  
2934 \testkern{LLL}{-}{170}{-}{RRR} \\ % lslash  
2935 \testkern{LLL}{-}{109}{-}{RRR} \\ % m  
2936 \testkern{LLL}{-}{110}{-}{RRR} \\ % n  
2937 \testkern{LLL}{-}{171}{-}{RRR} \\ % nacute  
2938 \testkern{LLL}{-}{172}{-}{RRR} \\ % ncaron  
2939 \testkern{LLL}{-}{241}{-}{RRR} \\ % ntilde  
2940 \testkern{LLL}{-}{111}{-}{RRR} \\ % o  
2941 \testkern{LLL}{-}{174}{-}{RRR} \\ % ohungarumlaut  
2942 \testkern{LLL}{-}{242}{-}{RRR} \\ % ograve  
2943 \testkern{LLL}{-}{243}{-}{RRR} \\ % oacute  
2944 \testkern{LLL}{-}{244}{-}{RRR} \\ % ocircumflex  
2945 \testkern{LLL}{-}{245}{-}{RRR} \\ % otilde  
2946 \testkern{LLL}{-}{246}{-}{RRR} \\ % odieresis  
2947 \testkern{LLL}{-}{247}{-}{RRR} \\ % oe  
2948 \testkern{LLL}{-}{248}{-}{RRR} \\ % oslash  
2949 \testkern{LLL}{-}{112}{-}{RRR} \\ % p  
2950 \testkern{LLL}{-}{113}{-}{RRR} \\ % q  
2951 \testkern{LLL}{-}{114}{-}{RRR} \\ % r  
2952 \testkern{LLL}{-}{175}{-}{RRR} \\ % racute  
2953 \testkern{LLL}{-}{176}{-}{RRR} \\ % rcaron  
2954 \testkern{LLL}{-}{115}{-}{RRR} \\ % s  
2955 \testkern{LLL}{-}{177}{-}{RRR} \\ % sacute  
2956 \testkern{LLL}{-}{178}{-}{RRR} \\ % scaron  
2957 \testkern{LLL}{-}{179}{-}{RRR} \\ % scedilla  
2958 \testkern{LLL}{-}{116}{-}{RRR} \\ % t  
2959 \testkern{LLL}{-}{180}{-}{RRR} \\ % tcaron  
2960 \testkern{LLL}{-}{181}{-}{RRR} \\ % tcedilla  
2961 \testkern{LLL}{-}{117}{-}{RRR} \\ % u  
2962 \testkern{LLL}{-}{182}{-}{RRR} \\ % uhungarumlaut  
2963 \testkern{LLL}{-}{183}{-}{RRR} \\ % uring  
2964 \testkern{LLL}{-}{249}{-}{RRR} \\ % ugrave  
2965 \testkern{LLL}{-}{250}{-}{RRR} \\ % uacute  
2966 \testkern{LLL}{-}{251}{-}{RRR} \\ % ucircumflex  
2967 \testkern{LLL}{-}{252}{-}{RRR} \\ % udieresis  
2968 \testkern{LLL}{-}{118}{-}{RRR} \\ % v  
2969 \testkern{LLL}{-}{119}{-}{RRR} \\ % w  
2970 \testkern{LLL}{-}{120}{-}{RRR} \\ % x  
2971 \testkern{LLL}{-}{121}{-}{RRR} \\ % y

```

2972 \testkern{LLL}{-}{184}{-}{RRR} \\ % ydieresis
2973 \testkern{LLL}{-}{253}{-}{RRR} \\ % yacute
2974 \testkern{LLL}{-}{122}{-}{RRR} \\ % z
2975 \testkern{LLL}{-}{185}{-}{RRR} \\ % zacute
2976 \testkern{LLL}{-}{186}{-}{RRR} \\ % zcaron
2977 \testkern{LLL}{-}{187}{-}{RRR} \\ % zdotaccent
2978 \testkern{LLL}{-}{240}{-}{RRR} \\ % eth
2979 \testkern{LLL}{-}{173}{-}{RRR} \\ % ng
2980 \testkern{LLL}{-}{254}{-}{RRR} \\ % thorn
2981 \testkern{LLL}{-}{255}{-}{RRR} \\ % germandbls
2982 \end{kerntable}
2983
2984 \end{document}
2985 </template & t1)

```

#### 6.4.2 TS1 encoding

```

2986 (*template & ts1)
2987 \listfiles
2988 %% Replace the 'XXX' in the next line by the 3- or 4-character long
2989 %% abbreviation for your font.
2990 \documentclass[family=XXX]{kerntest}
2991
2992 %% Replace the settings by these you want to test.
2993 \kernsetup{encoding=TS1,series=m,shape=n,example=hello}
2994 \kernsetup{size=14.40pt,baselineskip=16.5pt,papersize=a4paper}
2995
2996 %% The next line can be used to add a name suffix to the output |mtx| file.
2997 %% \kernsetup{extraname=normal}
2998
2999 %% If you are using a font with different design sizes and if you want
3000 %% to test one design size scaled to another one, you may input a
3001 %% modified fd file for your font. To generate this new fd file, just
3002 %% copy the original one, rename it, and modify the entries for the
3003 %% font shapes to use the design size you want to test.
3004 %% \input{t1XXX-1200.fd}
3005
3006 \begin{document}
3007
3008 %% This table of characters is sorted by similar glyphs, not by the
3009 %% encoding.
3010 %% Replace '{LLL}' and '{RRR}' in columns 1 resp. 5 by these glyphs
3011 %% that shall be tested.
3012 \begin{kerntable}
3013 \testkern{LLL}{-}{000}{-}{RRR} \\ % capitalgrave
3014 \testkern{LLL}{-}{001}{-}{RRR} \\ % capitalacute
3015 \testkern{LLL}{-}{002}{-}{RRR} \\ % capitalcircumflex
3016 \testkern{LLL}{-}{003}{-}{RRR} \\ % capitaltilde
3017 \testkern{LLL}{-}{004}{-}{RRR} \\ % capitaldieresis
3018 \testkern{LLL}{-}{005}{-}{RRR} \\ % capitalhungarumlaut
3019 \testkern{LLL}{-}{006}{-}{RRR} \\ % capitalring
3020 \testkern{LLL}{-}{007}{-}{RRR} \\ % capitalcaron
3021 \testkern{LLL}{-}{008}{-}{RRR} \\ % capitalbreve
3022 \testkern{LLL}{-}{009}{-}{RRR} \\ % capitalmacron
3023 \testkern{LLL}{-}{010}{-}{RRR} \\ % capitaldotaccent

```

3024 \testkern{LLL}{-}{011}{-}{RRR} \\ % cedilla  
3025 \testkern{LLL}{-}{012}{-}{RRR} \\ % ogonek  
3026 \testkern{LLL}{-}{013}{-}{RRR} \\ % quotesinglbase  
3027 \testkern{LLL}{-}{014}{-}{RRR} \\  
3028 \testkern{LLL}{-}{015}{-}{RRR} \\  
3029 \testkern{LLL}{-}{016}{-}{RRR} \\  
3030 \testkern{LLL}{-}{017}{-}{RRR} \\  
3031 \testkern{LLL}{-}{018}{-}{RRR} \\ % quotedblbase  
3032 \testkern{LLL}{-}{019}{-}{RRR} \\  
3033 \testkern{LLL}{-}{020}{-}{RRR} \\  
3034 \testkern{LLL}{-}{021}{-}{RRR} \\ % twelveudash  
3035 \testkern{LLL}{-}{022}{-}{RRR} \\ % threequartersemdash  
3036 \testkern{LLL}{-}{023}{-}{RRR} \\ % capitalcompwordmark  
3037 \testkern{LLL}{-}{024}{-}{RRR} \\ % arrowleft  
3038 \testkern{LLL}{-}{025}{-}{RRR} \\ % arrowright  
3039 \testkern{LLL}{-}{026}{-}{RRR} \\ % tieaccentlowercase  
3040 \testkern{LLL}{-}{027}{-}{RRR} \\ % tieaccentcapital  
3041 \testkern{LLL}{-}{028}{-}{RRR} \\ % newtieaccentlowercase  
3042 \testkern{LLL}{-}{029}{-}{RRR} \\ % newtieaccentcapital  
3043 \testkern{LLL}{-}{030}{-}{RRR} \\ % ascendercompwordmark  
3044 \testkern{LLL}{-}{031}{-}{RRR} \\ % blank  
3045 \testkern{LLL}{-}{032}{-}{RRR} \\  
3046 \testkern{LLL}{-}{033}{-}{RRR} \\  
3047 \testkern{LLL}{-}{034}{-}{RRR} \\  
3048 \testkern{LLL}{-}{035}{-}{RRR} \\  
3049 \testkern{LLL}{-}{036}{-}{RRR} \\ % dollar  
3050 \testkern{LLL}{-}{037}{-}{RRR} \\  
3051 \testkern{LLL}{-}{038}{-}{RRR} \\  
3052 \testkern{LLL}{-}{039}{-}{RRR} \\ % quotesingle  
3053 \testkern{LLL}{-}{040}{-}{RRR} \\  
3054 \testkern{LLL}{-}{041}{-}{RRR} \\  
3055 \testkern{LLL}{-}{042}{-}{RRR} \\ % asteriskcentered  
3056 \testkern{LLL}{-}{043}{-}{RRR} \\  
3057 \testkern{LLL}{-}{044}{-}{RRR} \\ % comma  
3058 \testkern{LLL}{-}{045}{-}{RRR} \\ % hyphendbl  
3059 \testkern{LLL}{-}{046}{-}{RRR} \\ % period  
3060 \testkern{LLL}{-}{047}{-}{RRR} \\ % fraction  
3061 \testkern{LLL}{-}{048}{-}{RRR} \\ % zerooldstyle  
3062 \testkern{LLL}{-}{049}{-}{RRR} \\ % oneoldstyle  
3063 \testkern{LLL}{-}{050}{-}{RRR} \\ % twooldstyle  
3064 \testkern{LLL}{-}{051}{-}{RRR} \\ % threeoldstyle  
3065 \testkern{LLL}{-}{052}{-}{RRR} \\ % fouroldstyle  
3066 \testkern{LLL}{-}{053}{-}{RRR} \\ % fiveoldstyle  
3067 \testkern{LLL}{-}{054}{-}{RRR} \\ % sixoldstyle  
3068 \testkern{LLL}{-}{055}{-}{RRR} \\ % sevenoldstyle  
3069 \testkern{LLL}{-}{056}{-}{RRR} \\ % eightoldstyle  
3070 \testkern{LLL}{-}{057}{-}{RRR} \\ % nineoldstyle  
3071 \testkern{LLL}{-}{058}{-}{RRR} \\  
3072 \testkern{LLL}{-}{059}{-}{RRR} \\  
3073 \testkern{LLL}{-}{060}{-}{RRR} \\ % angbracketleft  
3074 \testkern{LLL}{-}{061}{-}{RRR} \\ % minus  
3075 \testkern{LLL}{-}{062}{-}{RRR} \\ % angbracketright  
3076 \testkern{LLL}{-}{063}{-}{RRR} \\  
3077 \testkern{LLL}{-}{064}{-}{RRR} \\

3078 \testkern{LLL}{-}{065}{-}{RRR} \\
3079 \testkern{LLL}{-}{066}{-}{RRR} \\
3080 \testkern{LLL}{-}{067}{-}{RRR} \\
3081 \testkern{LLL}{-}{068}{-}{RRR} \\
3082 \testkern{LLL}{-}{069}{-}{RRR} \\
3083 \testkern{LLL}{-}{070}{-}{RRR} \\
3084 \testkern{LLL}{-}{071}{-}{RRR} \\
3085 \testkern{LLL}{-}{072}{-}{RRR} \\
3086 \testkern{LLL}{-}{073}{-}{RRR} \\
3087 \testkern{LLL}{-}{074}{-}{RRR} \\
3088 \testkern{LLL}{-}{075}{-}{RRR} \\
3089 \testkern{LLL}{-}{076}{-}{RRR} \\
3090 \testkern{LLL}{-}{077}{-}{RRR} \\ \% Omegainv
3091 \testkern{LLL}{-}{078}{-}{RRR} \\
3092 \testkern{LLL}{-}{079}{-}{RRR} \\ \% bigcircle
3093 \testkern{LLL}{-}{080}{-}{RRR} \\
3094 \testkern{LLL}{-}{081}{-}{RRR} \\
3095 \testkern{LLL}{-}{082}{-}{RRR} \\
3096 \testkern{LLL}{-}{083}{-}{RRR} \\
3097 \testkern{LLL}{-}{084}{-}{RRR} \\
3098 \testkern{LLL}{-}{085}{-}{RRR} \\
3099 \testkern{LLL}{-}{086}{-}{RRR} \\
3100 \testkern{LLL}{-}{087}{-}{RRR} \\ \% Omega
3101 \testkern{LLL}{-}{088}{-}{RRR} \\
3102 \testkern{LLL}{-}{089}{-}{RRR} \\
3103 \testkern{LLL}{-}{090}{-}{RRR} \\
3104 \testkern{LLL}{-}{091}{-}{RRR} \\ \% openbracketleft
3105 \testkern{LLL}{-}{092}{-}{RRR} \\
3106 \testkern{LLL}{-}{093}{-}{RRR} \\ \% openbracketright
3107 \testkern{LLL}{-}{094}{-}{RRR} \\ \% arrowup
3108 \testkern{LLL}{-}{095}{-}{RRR} \\ \% arrowdown
3109 \testkern{LLL}{-}{096}{-}{RRR} \\ \% asciigrave
3110 \testkern{LLL}{-}{097}{-}{RRR} \\
3111 \testkern{LLL}{-}{098}{-}{RRR} \\ \% born
3112 \testkern{LLL}{-}{099}{-}{RRR} \\ \% divorced
3113 \testkern{LLL}{-}{100}{-}{RRR} \\ \% died
3114 \testkern{LLL}{-}{101}{-}{RRR} \\
3115 \testkern{LLL}{-}{102}{-}{RRR} \\
3116 \testkern{LLL}{-}{103}{-}{RRR} \\
3117 \testkern{LLL}{-}{104}{-}{RRR} \\
3118 \testkern{LLL}{-}{105}{-}{RRR} \\
3119 \testkern{LLL}{-}{106}{-}{RRR} \\
3120 \testkern{LLL}{-}{107}{-}{RRR} \\
3121 \testkern{LLL}{-}{108}{-}{RRR} \\ \% leaf
3122 \testkern{LLL}{-}{109}{-}{RRR} \\ \% married
3123 \testkern{LLL}{-}{110}{-}{RRR} \\ \% musicalnote
3124 \testkern{LLL}{-}{111}{-}{RRR} \\
3125 \testkern{LLL}{-}{112}{-}{RRR} \\
3126 \testkern{LLL}{-}{113}{-}{RRR} \\
3127 \testkern{LLL}{-}{114}{-}{RRR} \\
3128 \testkern{LLL}{-}{115}{-}{RRR} \\
3129 \testkern{LLL}{-}{116}{-}{RRR} \\
3130 \testkern{LLL}{-}{117}{-}{RRR} \\
3131 \testkern{LLL}{-}{118}{-}{RRR} \\

3132 \testkern{LLL}{-}{119}{-}{RRR} \\
3133 \testkern{LLL}{-}{120}{-}{RRR} \\
3134 \testkern{LLL}{-}{121}{-}{RRR} \\
3135 \testkern{LLL}{-}{122}{-}{RRR} \\
3136 \testkern{LLL}{-}{123}{-}{RRR} \\
3137 \testkern{LLL}{-}{124}{-}{RRR} \\
3138 \testkern{LLL}{-}{125}{-}{RRR} \\
3139 \testkern{LLL}{-}{126}{-}{RRR} \\ \ % tildelow
3140 \testkern{LLL}{-}{127}{-}{RRR} \\ \ % hyphendblchar
3141 \testkern{LLL}{-}{128}{-}{RRR} \\ \ % asciibreve
3142 \testkern{LLL}{-}{129}{-}{RRR} \\ \ % asciicaron
3143 \testkern{LLL}{-}{130}{-}{RRR} \\ \ % asciiacutedbl
3144 \testkern{LLL}{-}{131}{-}{RRR} \\ \ % asciigravedbl
3145 \testkern{LLL}{-}{132}{-}{RRR} \\ \ % dagger
3146 \testkern{LLL}{-}{133}{-}{RRR} \\ \ % daggerdbl
3147 \testkern{LLL}{-}{134}{-}{RRR} \\ \ % bardbl
3148 \testkern{LLL}{-}{135}{-}{RRR} \\ \ % perthousand
3149 \testkern{LLL}{-}{136}{-}{RRR} \\ \ % bullet
3150 \testkern{LLL}{-}{137}{-}{RRR} \\ \ % centigrade
3151 \testkern{LLL}{-}{138}{-}{RRR} \\ \ % dollaroldstyle
3152 \testkern{LLL}{-}{139}{-}{RRR} \\ \ % centoldstyle
3153 \testkern{LLL}{-}{140}{-}{RRR} \\ \ % florin
3154 \testkern{LLL}{-}{141}{-}{RRR} \\ \ % colonmonetary
3155 \testkern{LLL}{-}{142}{-}{RRR} \\ \ % won
3156 \testkern{LLL}{-}{143}{-}{RRR} \\ \ % naira
3157 \testkern{LLL}{-}{144}{-}{RRR} \\ \ % guarani
3158 \testkern{LLL}{-}{145}{-}{RRR} \\ \ % peso
3159 \testkern{LLL}{-}{146}{-}{RRR} \\ \ % lira
3160 \testkern{LLL}{-}{147}{-}{RRR} \\ \ % recipe
3161 \testkern{LLL}{-}{148}{-}{RRR} \\ \ % interrobang
3162 \testkern{LLL}{-}{149}{-}{RRR} \\ \ % interrobangdown
3163 \testkern{LLL}{-}{150}{-}{RRR} \\ \ % dong
3164 \testkern{LLL}{-}{151}{-}{RRR} \\ \ % trademark
3165 \testkern{LLL}{-}{152}{-}{RRR} \\ \ % pertenthousand
3166 \testkern{LLL}{-}{153}{-}{RRR} \\ \ % pilcrow
3167 \testkern{LLL}{-}{154}{-}{RRR} \\ \ % baht
3168 \testkern{LLL}{-}{155}{-}{RRR} \\ \ % numero
3169 \testkern{LLL}{-}{156}{-}{RRR} \\ \ % discount
3170 \testkern{LLL}{-}{157}{-}{RRR} \\ \ % estimated
3171 \testkern{LLL}{-}{158}{-}{RRR} \\ \ % openbullet
3172 \testkern{LLL}{-}{159}{-}{RRR} \\ \ % servicemark
3173 \testkern{LLL}{-}{160}{-}{RRR} \\ \ % quillbracketleft
3174 \testkern{LLL}{-}{161}{-}{RRR} \\ \ % quillbracketright
3175 \testkern{LLL}{-}{162}{-}{RRR} \\ \ % cent
3176 \testkern{LLL}{-}{163}{-}{RRR} \\ \ % sterling
3177 \testkern{LLL}{-}{164}{-}{RRR} \\ \ % currency
3178 \testkern{LLL}{-}{165}{-}{RRR} \\ \ % yen
3179 \testkern{LLL}{-}{166}{-}{RRR} \\ \ % brokenbar
3180 \testkern{LLL}{-}{167}{-}{RRR} \\ \ % section
3181 \testkern{LLL}{-}{168}{-}{RRR} \\ \ % asciidieresis
3182 \testkern{LLL}{-}{169}{-}{RRR} \\ \ % copyright
3183 \testkern{LLL}{-}{170}{-}{RRR} \\ \ % ordfeminine
3184 \testkern{LLL}{-}{171}{-}{RRR} \\ \ % copleft
3185 \testkern{LLL}{-}{172}{-}{RRR} \\ \ % logicalnot

3186 \testkern{LLL}{-}{173}{-}{RRR} \\ % circledP  
3187 \testkern{LLL}{-}{174}{-}{RRR} \\ % registered  
3188 \testkern{LLL}{-}{175}{-}{RRR} \\ % asciimacron  
3189 \testkern{LLL}{-}{176}{-}{RRR} \\ % degree  
3190 \testkern{LLL}{-}{177}{-}{RRR} \\ % plusminus  
3191 \testkern{LLL}{-}{178}{-}{RRR} \\ % twosuperior  
3192 \testkern{LLL}{-}{179}{-}{RRR} \\ % threesuperior  
3193 \testkern{LLL}{-}{180}{-}{RRR} \\ % asciiacute  
3194 \testkern{LLL}{-}{181}{-}{RRR} \\ % mu  
3195 \testkern{LLL}{-}{182}{-}{RRR} \\ % paragraph  
3196 \testkern{LLL}{-}{183}{-}{RRR} \\ % periodcentered  
3197 \testkern{LLL}{-}{184}{-}{RRR} \\ % referencemark  
3198 \testkern{LLL}{-}{185}{-}{RRR} \\ % onesuperior  
3199 \testkern{LLL}{-}{186}{-}{RRR} \\ % ordmasculine  
3200 \testkern{LLL}{-}{187}{-}{RRR} \\ % radical  
3201 \testkern{LLL}{-}{188}{-}{RRR} \\ % onequarter  
3202 \testkern{LLL}{-}{189}{-}{RRR} \\ % onehalf  
3203 \testkern{LLL}{-}{190}{-}{RRR} \\ % threequarters  
3204 \testkern{LLL}{-}{191}{-}{RRR} \\ % euro  
3205 \testkern{LLL}{-}{192}{-}{RRR} \\  
3206 \testkern{LLL}{-}{193}{-}{RRR} \\  
3207 \testkern{LLL}{-}{194}{-}{RRR} \\  
3208 \testkern{LLL}{-}{195}{-}{RRR} \\  
3209 \testkern{LLL}{-}{196}{-}{RRR} \\  
3210 \testkern{LLL}{-}{197}{-}{RRR} \\  
3211 \testkern{LLL}{-}{198}{-}{RRR} \\  
3212 \testkern{LLL}{-}{199}{-}{RRR} \\  
3213 \testkern{LLL}{-}{200}{-}{RRR} \\  
3214 \testkern{LLL}{-}{201}{-}{RRR} \\  
3215 \testkern{LLL}{-}{202}{-}{RRR} \\  
3216 \testkern{LLL}{-}{203}{-}{RRR} \\  
3217 \testkern{LLL}{-}{204}{-}{RRR} \\  
3218 \testkern{LLL}{-}{205}{-}{RRR} \\  
3219 \testkern{LLL}{-}{206}{-}{RRR} \\  
3220 \testkern{LLL}{-}{207}{-}{RRR} \\  
3221 \testkern{LLL}{-}{208}{-}{RRR} \\  
3222 \testkern{LLL}{-}{209}{-}{RRR} \\  
3223 \testkern{LLL}{-}{210}{-}{RRR} \\  
3224 \testkern{LLL}{-}{211}{-}{RRR} \\  
3225 \testkern{LLL}{-}{212}{-}{RRR} \\  
3226 \testkern{LLL}{-}{213}{-}{RRR} \\  
3227 \testkern{LLL}{-}{214}{-}{RRR} \\ % multiply  
3228 \testkern{LLL}{-}{215}{-}{RRR} \\  
3229 \testkern{LLL}{-}{216}{-}{RRR} \\  
3230 \testkern{LLL}{-}{217}{-}{RRR} \\  
3231 \testkern{LLL}{-}{218}{-}{RRR} \\  
3232 \testkern{LLL}{-}{219}{-}{RRR} \\  
3233 \testkern{LLL}{-}{220}{-}{RRR} \\  
3234 \testkern{LLL}{-}{221}{-}{RRR} \\  
3235 \testkern{LLL}{-}{222}{-}{RRR} \\  
3236 \testkern{LLL}{-}{223}{-}{RRR} \\  
3237 \testkern{LLL}{-}{224}{-}{RRR} \\  
3238 \testkern{LLL}{-}{225}{-}{RRR} \\  
3239 \testkern{LLL}{-}{226}{-}{RRR} \\

```

3240 \testkern{LLL}{-}{227}{-}{RRR} \\
3241 \testkern{LLL}{-}{228}{-}{RRR} \\
3242 \testkern{LLL}{-}{229}{-}{RRR} \\
3243 \testkern{LLL}{-}{230}{-}{RRR} \\
3244 \testkern{LLL}{-}{231}{-}{RRR} \\
3245 \testkern{LLL}{-}{232}{-}{RRR} \\
3246 \testkern{LLL}{-}{233}{-}{RRR} \\
3247 \testkern{LLL}{-}{234}{-}{RRR} \\
3248 \testkern{LLL}{-}{235}{-}{RRR} \\
3249 \testkern{LLL}{-}{236}{-}{RRR} \\
3250 \testkern{LLL}{-}{237}{-}{RRR} \\
3251 \testkern{LLL}{-}{238}{-}{RRR} \\
3252 \testkern{LLL}{-}{239}{-}{RRR} \\
3253 \testkern{LLL}{-}{240}{-}{RRR} \\
3254 \testkern{LLL}{-}{241}{-}{RRR} \\
3255 \testkern{LLL}{-}{242}{-}{RRR} \\
3256 \testkern{LLL}{-}{243}{-}{RRR} \\
3257 \testkern{LLL}{-}{244}{-}{RRR} \\
3258 \testkern{LLL}{-}{245}{-}{RRR} \\
3259 \testkern{LLL}{-}{246}{-}{RRR} \\ % divide
3260 \testkern{LLL}{-}{247}{-}{RRR} \\
3261 \testkern{LLL}{-}{248}{-}{RRR} \\
3262 \testkern{LLL}{-}{249}{-}{RRR} \\
3263 \testkern{LLL}{-}{250}{-}{RRR} \\
3264 \testkern{LLL}{-}{251}{-}{RRR} \\
3265 \testkern{LLL}{-}{252}{-}{RRR} \\
3266 \testkern{LLL}{-}{253}{-}{RRR} \\
3267 \testkern{LLL}{-}{254}{-}{RRR} \\
3268 \testkern{LLL}{-}{255}{-}{RRR} \\
3269 \end{kerntable}
3270
3271 \end{document}
3272 </template & ts1)

```

### 6.4.3 OT1 encoding

```

3273 <*template & ot1)
3274 \listfiles
3275 %% Replace the 'XXX' in the next line by the 3- or 4-character long
3276 %% abbreviation for your font.
3277 \documentclass[family=XXX]{kerntest}
3278
3279 %% Replace the settings by these you want to test.
3280 \kernsetup{encoding=OT1,series=m,shape=n,example=hello}
3281 \kernsetup{size=14.40pt,baselineskip=16.5pt,papersize=a4paper}
3282
3283 %% The next line can be used to add a name suffix to the output mtx file.
3284 %% \kernsetup{extraname=normal}
3285
3286 %% Set encoding parameters.
3287 %% Set ligaturing: 2=all, 1=some, 0=none
3288 %% \encodingsetup{ligaturing=2}
3289 %% Normal: dollar, Italic: sterling
3290 %% \encodingsetup{italicizing=false}
3291

```

```

3292 %% If you are using a font with different design sizes and if you want
3293 %% to test one design size scaled to another one, you may input a
3294 %% modified fd file for your font. To generate this new fd file, just
3295 %% copy the original one, rename it, and modify the entries for the
3296 %% font shapes to use the design size you want to test.
3297 %% \input{ot1XXX-1200.fd}
3298
3299 %% The following lines show some possible glyph classes. You should
3300 %% add all classes you need.
3301 \newglyphclass{right}{E}{E,AE,OE}
3302
3303 \newglyphclass{left}{H}{B,D,F,H,I,J,K,L,N,P,R}
3304
3305 \newglyphclass{right}{H}{H,I,J,N}
3306
3307 \newglyphclass{right}{S}{S,dollar}
3308 \newglyphclass{left}{S}{S,dollar}
3309
3310 \newglyphclass{left}{a}{a,ae}
3311 \newglyphclass{right}{e}{e,ae,oe}
3312
3313 \newglyphclass{left}{f}{f,ff,fi,fl,ffi,ffl}
3314 \newglyphclass{right}{f}{f,ff}
3315
3316 \newglyphclass{right}{i}{i,fi,ffi,dotlessi}
3317 \newglyphclass{left}{i}{i,dotlessi}
3318
3319 \newglyphclass{right}{l}{fl,ffl,l}
3320
3321 \newglyphclass{right}{m}{m,n}
3322 \newglyphclass{left}{m}{m,n}
3323
3324 \newglyphclass{left}{o}{o,oe}
3325
3326 \begin{document}
3327
3328 %% This table of characters is sorted by similar glyphs, not by the
3329 %% encoding.
3330 %% Replace '{LLL}' and '{RRR}' in columns 1 resp. 5 by these glyphs
3331 %% that shall be tested.
3332 \begin{kerntable}
3333 \testkern{LLL}{-}{018}{-}{RRR} \ \ % grave
3334 \testkern{LLL}{-}{019}{-}{RRR} \ \ % acute
3335 \testkern{LLL}{-}{094}{-}{RRR} \ \ % circumflex
3336 \testkern{LLL}{-}{126}{-}{RRR} \ \ % tilde
3337 \testkern{LLL}{-}{127}{-}{RRR} \ \ % dieresis
3338 \testkern{LLL}{-}{125}{-}{RRR} \ \ % hungarumlaut/braceright
3339 \testkern{LLL}{-}{023}{-}{RRR} \ \ % ringfitted
3340 \testkern{LLL}{-}{020}{-}{RRR} \ \ % caron
3341 \testkern{LLL}{-}{021}{-}{RRR} \ \ % breve
3342 \testkern{LLL}{-}{022}{-}{RRR} \ \ % macron
3343 \testkern{LLL}{-}{095}{-}{RRR} \ \ % dotaccent/underscore
3344 \testkern{LLL}{-}{024}{-}{RRR} \ \ % cedilla
3345 \testkern{LLL}{-}{042}{-}{RRR} \ \ % asterisk

```

3346 \testkern{LLL}{-}{032}{-}{RRR} \\ % lslashslash/visiblespace  
3347 \testkern{LLL}{-}{096}{-}{RRR} \\ % quoteleft  
3348 \testkern{LLL}{-}{039}{-}{RRR} \\ % quoteright  
3349 \testkern{LLL}{-}{092}{-}{RRR} \\ % quotedblleft  
3350 \testkern{LLL}{-}{034}{-}{RRR} \\ % quotedblright/quotedbl  
3351 \testkern{LLL}{-}{046}{-}{RRR} \\ % period  
3352 \testkern{LLL}{-}{044}{-}{RRR} \\ % comma  
3353 \testkern{LLL}{-}{058}{-}{RRR} \\ % colon  
3354 \testkern{LLL}{-}{059}{-}{RRR} \\ % semicolon  
3355 \testkern{LLL}{-}{033}{-}{RRR} \\ % exclam  
3356 \testkern{LLL}{-}{063}{-}{RRR} \\ % question  
3357 \testkern{LLL}{-}{060}{-}{RRR} \\ % exclamdown/less  
3358 \testkern{LLL}{-}{062}{-}{RRR} \\ % questiondown/greater  
3359 \testkern{LLL}{-}{045}{-}{RRR} \\ % hyphen  
3360 \testkern{LLL}{-}{123}{-}{RRR} \\ % rangedash/braceleft  
3361 \testkern{LLL}{-}{124}{-}{RRR} \\ % punctdash/bar  
3362 \testkern{LLL}{-}{043}{-}{RRR} \\ % plus  
3363 \testkern{LLL}{-}{061}{-}{RRR} \\ % equal  
3364 \testkern{LLL}{-}{047}{-}{RRR} \\ % slash  
3365 \testkern{LLL}{-}{040}{-}{RRR} \\ % parenleft  
3366 \testkern{LLL}{-}{041}{-}{RRR} \\ % parenright  
3367 \testkern{LLL}{-}{091}{-}{RRR} \\ % bracketleft  
3368 \testkern{LLL}{-}{093}{-}{RRR} \\ % bracketright  
3369 \testkern{LLL}{-}{035}{-}{RRR} \\ % numbersign  
3370 \testkern{LLL}{-}{037}{-}{RRR} \\ % percent  
3371 \testkern{LLL}{-}{038}{-}{RRR} \\ % ampersand  
3372 \testkern{LLL}{-}{064}{-}{RRR} \\ % at  
3373 \testkern{LLL}{-}{048}{-}{RRR} \\ % zero  
3374 \testkern{LLL}{-}{049}{-}{RRR} \\ % one  
3375 \testkern{LLL}{-}{050}{-}{RRR} \\ % two  
3376 \testkern{LLL}{-}{051}{-}{RRR} \\ % three  
3377 \testkern{LLL}{-}{052}{-}{RRR} \\ % four  
3378 \testkern{LLL}{-}{053}{-}{RRR} \\ % five  
3379 \testkern{LLL}{-}{054}{-}{RRR} \\ % six  
3380 \testkern{LLL}{-}{055}{-}{RRR} \\ % seven  
3381 \testkern{LLL}{-}{056}{-}{RRR} \\ % eight  
3382 \testkern{LLL}{-}{057}{-}{RRR} \\ % nine  
3383 \testkern{LLL}{-}{065}{-}{RRR} \\ % A  
3384 \testkern{LLL}{-}{029}{-}{RRR} \\ % AE  
3385 \testkern{LLL}{-}{066}{-}{RRR} \\ % B  
3386 \testkern{LLL}{-}{067}{-}{RRR} \\ % C  
3387 \testkern{LLL}{-}{068}{-}{RRR} \\ % D  
3388 \testkern{LLL}{-}{069}{-}{RRR} \\ % E  
3389 \testkern{LLL}{-}{070}{-}{RRR} \\ % F  
3390 \testkern{LLL}{-}{071}{-}{RRR} \\ % G  
3391 \testkern{LLL}{-}{072}{-}{RRR} \\ % H  
3392 \testkern{LLL}{-}{073}{-}{RRR} \\ % I  
3393 \testkern{LLL}{-}{074}{-}{RRR} \\ % J  
3394 \testkern{LLL}{-}{075}{-}{RRR} \\ % K  
3395 \testkern{LLL}{-}{076}{-}{RRR} \\ % L  
3396 \testkern{LLL}{-}{138}{-}{RRR} \\ % Lslash  
3397 \testkern{LLL}{-}{077}{-}{RRR} \\ % M  
3398 \testkern{LLL}{-}{078}{-}{RRR} \\ % N  
3399 \testkern{LLL}{-}{079}{-}{RRR} \\ % O

3400 \testkern{LLL}{-}{030}{-}{RRR} \\ % OE  
3401 \testkern{LLL}{-}{031}{-}{RRR} \\ % Oslash  
3402 \testkern{LLL}{-}{080}{-}{RRR} \\ % P  
3403 \testkern{LLL}{-}{081}{-}{RRR} \\ % Q  
3404 \testkern{LLL}{-}{082}{-}{RRR} \\ % R  
3405 \testkern{LLL}{-}{083}{-}{RRR} \\ % S  
3406 \testkern{LLL}{-}{036}{-}{RRR} \\ % dollar/sterling  
3407 \testkern{LLL}{-}{084}{-}{RRR} \\ % T  
3408 \testkern{LLL}{-}{085}{-}{RRR} \\ % U  
3409 \testkern{LLL}{-}{086}{-}{RRR} \\ % V  
3410 \testkern{LLL}{-}{087}{-}{RRR} \\ % W  
3411 \testkern{LLL}{-}{088}{-}{RRR} \\ % X  
3412 \testkern{LLL}{-}{089}{-}{RRR} \\ % Y  
3413 \testkern{LLL}{-}{090}{-}{RRR} \\ % Z  
3414 \testkern{LLL}{-}{097}{-}{RRR} \\ % a  
3415 \testkern{LLL}{-}{026}{-}{RRR} \\ % ae  
3416 \testkern{LLL}{-}{098}{-}{RRR} \\ % b  
3417 \testkern{LLL}{-}{099}{-}{RRR} \\ % c  
3418 \testkern{LLL}{-}{100}{-}{RRR} \\ % d  
3419 \testkern{LLL}{-}{101}{-}{RRR} \\ % e  
3420 \testkern{LLL}{-}{102}{-}{RRR} \\ % f  
3421 \testkern{LLL}{-}{011}{-}{RRR} \\ % ff/arrowup  
3422 \testkern{LLL}{-}{012}{-}{RRR} \\ % fi/arrowdown  
3423 \testkern{LLL}{-}{013}{-}{RRR} \\ % fl/quotesingle  
3424 \testkern{LLL}{-}{014}{-}{RRR} \\ % ffi/exclamdown  
3425 \testkern{LLL}{-}{015}{-}{RRR} \\ % ffl/questiondown  
3426 \testkern{LLL}{-}{103}{-}{RRR} \\ % g  
3427 \testkern{LLL}{-}{104}{-}{RRR} \\ % h  
3428 \testkern{LLL}{-}{105}{-}{RRR} \\ % i  
3429 \testkern{LLL}{-}{016}{-}{RRR} \\ % dotlessi  
3430 \testkern{LLL}{-}{025}{-}{RRR} \\ % germandbls  
3431 \testkern{LLL}{-}{106}{-}{RRR} \\ % j  
3432 \testkern{LLL}{-}{017}{-}{RRR} \\ % dotlessj  
3433 \testkern{LLL}{-}{107}{-}{RRR} \\ % k  
3434 \testkern{LLL}{-}{108}{-}{RRR} \\ % l  
3435 \testkern{LLL}{-}{170}{-}{RRR} \\ % lslash  
3436 \testkern{LLL}{-}{109}{-}{RRR} \\ % m  
3437 \testkern{LLL}{-}{110}{-}{RRR} \\ % n  
3438 \testkern{LLL}{-}{111}{-}{RRR} \\ % o  
3439 \testkern{LLL}{-}{027}{-}{RRR} \\ % oe  
3440 \testkern{LLL}{-}{028}{-}{RRR} \\ % oslash  
3441 \testkern{LLL}{-}{112}{-}{RRR} \\ % p  
3442 \testkern{LLL}{-}{113}{-}{RRR} \\ % q  
3443 \testkern{LLL}{-}{114}{-}{RRR} \\ % r  
3444 \testkern{LLL}{-}{115}{-}{RRR} \\ % s  
3445 \testkern{LLL}{-}{116}{-}{RRR} \\ % t  
3446 \testkern{LLL}{-}{117}{-}{RRR} \\ % u  
3447 \testkern{LLL}{-}{118}{-}{RRR} \\ % v  
3448 \testkern{LLL}{-}{119}{-}{RRR} \\ % w  
3449 \testkern{LLL}{-}{120}{-}{RRR} \\ % x  
3450 \testkern{LLL}{-}{121}{-}{RRR} \\ % y  
3451 \testkern{LLL}{-}{122}{-}{RRR} \\ % z  
3452 \testkern{LLL}{-}{000}{-}{RRR} \\ % Gamma  
3453 \testkern{LLL}{-}{001}{-}{RRR} \\ % Delta

```

3454 \testkern{LLL}{-}{002}{-}{RRR} \\ % Theta
3455 \testkern{LLL}{-}{003}{-}{RRR} \\ % Lambda
3456 \testkern{LLL}{-}{004}{-}{RRR} \\ % Xi
3457 \testkern{LLL}{-}{005}{-}{RRR} \\ % Pi
3458 \testkern{LLL}{-}{006}{-}{RRR} \\ % Sigma
3459 \testkern{LLL}{-}{007}{-}{RRR} \\ % Upsilon
3460 \testkern{LLL}{-}{008}{-}{RRR} \\ % Phi
3461 \testkern{LLL}{-}{009}{-}{RRR} \\ % Psi
3462 \testkern{LLL}{-}{010}{-}{RRR} \\ % Omega
3463 \iffalse
3464 \testkern{LLL}{-}{128}{-}{RRR} \\ % .notdef.
3465 \testkern{LLL}{-}{129}{-}{RRR} \\ % .notdef.
3466 \testkern{LLL}{-}{130}{-}{RRR} \\ % .notdef.
3467 \testkern{LLL}{-}{131}{-}{RRR} \\ % .notdef.
3468 \testkern{LLL}{-}{132}{-}{RRR} \\ % .notdef.
3469 \testkern{LLL}{-}{133}{-}{RRR} \\ % .notdef.
3470 \testkern{LLL}{-}{134}{-}{RRR} \\ % .notdef.
3471 \testkern{LLL}{-}{135}{-}{RRR} \\ % .notdef.
3472 \testkern{LLL}{-}{136}{-}{RRR} \\ % .notdef.
3473 \testkern{LLL}{-}{137}{-}{RRR} \\ % .notdef.
3474 \testkern{LLL}{-}{139}{-}{RRR} \\ % .notdef.
3475 \testkern{LLL}{-}{140}{-}{RRR} \\ % .notdef.
3476 \testkern{LLL}{-}{141}{-}{RRR} \\ % .notdef.
3477 \testkern{LLL}{-}{142}{-}{RRR} \\ % .notdef.
3478 \testkern{LLL}{-}{143}{-}{RRR} \\ % .notdef.
3479 \testkern{LLL}{-}{144}{-}{RRR} \\ % .notdef.
3480 \testkern{LLL}{-}{145}{-}{RRR} \\ % .notdef.
3481 \testkern{LLL}{-}{146}{-}{RRR} \\ % .notdef.
3482 \testkern{LLL}{-}{147}{-}{RRR} \\ % .notdef.
3483 \testkern{LLL}{-}{148}{-}{RRR} \\ % .notdef.
3484 \testkern{LLL}{-}{149}{-}{RRR} \\ % .notdef.
3485 \testkern{LLL}{-}{150}{-}{RRR} \\ % .notdef.
3486 \testkern{LLL}{-}{151}{-}{RRR} \\ % .notdef.
3487 \testkern{LLL}{-}{152}{-}{RRR} \\ % .notdef.
3488 \testkern{LLL}{-}{153}{-}{RRR} \\ % .notdef.
3489 \testkern{LLL}{-}{154}{-}{RRR} \\ % .notdef.
3490 \testkern{LLL}{-}{155}{-}{RRR} \\ % .notdef.
3491 \testkern{LLL}{-}{156}{-}{RRR} \\ % .notdef.
3492 \testkern{LLL}{-}{157}{-}{RRR} \\ % .notdef.
3493 \testkern{LLL}{-}{158}{-}{RRR} \\ % .notdef.
3494 \testkern{LLL}{-}{159}{-}{RRR} \\ % .notdef.
3495 \testkern{LLL}{-}{160}{-}{RRR} \\ % .notdef.
3496 \testkern{LLL}{-}{161}{-}{RRR} \\ % .notdef.
3497 \testkern{LLL}{-}{162}{-}{RRR} \\ % .notdef.
3498 \testkern{LLL}{-}{163}{-}{RRR} \\ % .notdef.
3499 \testkern{LLL}{-}{164}{-}{RRR} \\ % .notdef.
3500 \testkern{LLL}{-}{165}{-}{RRR} \\ % .notdef.
3501 \testkern{LLL}{-}{166}{-}{RRR} \\ % .notdef.
3502 \testkern{LLL}{-}{167}{-}{RRR} \\ % .notdef.
3503 \testkern{LLL}{-}{168}{-}{RRR} \\ % .notdef.
3504 \testkern{LLL}{-}{169}{-}{RRR} \\ % .notdef.
3505 \testkern{LLL}{-}{171}{-}{RRR} \\ % .notdef.
3506 \testkern{LLL}{-}{172}{-}{RRR} \\ % .notdef.
3507 \testkern{LLL}{-}{173}{-}{RRR} \\ % .notdef.

```

3508 \testkern{LLL}{-}{174}{-}{RRR} \\ % .notdef.  
3509 \testkern{LLL}{-}{175}{-}{RRR} \\ % .notdef.  
3510 \testkern{LLL}{-}{176}{-}{RRR} \\ % .notdef.  
3511 \testkern{LLL}{-}{177}{-}{RRR} \\ % .notdef.  
3512 \testkern{LLL}{-}{178}{-}{RRR} \\ % .notdef.  
3513 \testkern{LLL}{-}{179}{-}{RRR} \\ % .notdef.  
3514 \testkern{LLL}{-}{180}{-}{RRR} \\ % .notdef.  
3515 \testkern{LLL}{-}{181}{-}{RRR} \\ % .notdef.  
3516 \testkern{LLL}{-}{182}{-}{RRR} \\ % .notdef.  
3517 \testkern{LLL}{-}{183}{-}{RRR} \\ % .notdef.  
3518 \testkern{LLL}{-}{184}{-}{RRR} \\ % .notdef.  
3519 \testkern{LLL}{-}{185}{-}{RRR} \\ % .notdef.  
3520 \testkern{LLL}{-}{186}{-}{RRR} \\ % .notdef.  
3521 \testkern{LLL}{-}{187}{-}{RRR} \\ % .notdef.  
3522 \testkern{LLL}{-}{188}{-}{RRR} \\ % .notdef.  
3523 \testkern{LLL}{-}{189}{-}{RRR} \\ % .notdef.  
3524 \testkern{LLL}{-}{190}{-}{RRR} \\ % .notdef.  
3525 \testkern{LLL}{-}{191}{-}{RRR} \\ % .notdef.  
3526 \testkern{LLL}{-}{192}{-}{RRR} \\ % .notdef.  
3527 \testkern{LLL}{-}{193}{-}{RRR} \\ % .notdef.  
3528 \testkern{LLL}{-}{194}{-}{RRR} \\ % .notdef.  
3529 \testkern{LLL}{-}{195}{-}{RRR} \\ % .notdef.  
3530 \testkern{LLL}{-}{196}{-}{RRR} \\ % .notdef.  
3531 \testkern{LLL}{-}{197}{-}{RRR} \\ % .notdef.  
3532 \testkern{LLL}{-}{198}{-}{RRR} \\ % .notdef.  
3533 \testkern{LLL}{-}{199}{-}{RRR} \\ % .notdef.  
3534 \testkern{LLL}{-}{200}{-}{RRR} \\ % .notdef.  
3535 \testkern{LLL}{-}{201}{-}{RRR} \\ % .notdef.  
3536 \testkern{LLL}{-}{202}{-}{RRR} \\ % .notdef.  
3537 \testkern{LLL}{-}{203}{-}{RRR} \\ % .notdef.  
3538 \testkern{LLL}{-}{204}{-}{RRR} \\ % .notdef.  
3539 \testkern{LLL}{-}{205}{-}{RRR} \\ % .notdef.  
3540 \testkern{LLL}{-}{206}{-}{RRR} \\ % .notdef.  
3541 \testkern{LLL}{-}{207}{-}{RRR} \\ % .notdef.  
3542 \testkern{LLL}{-}{208}{-}{RRR} \\ % .notdef.  
3543 \testkern{LLL}{-}{209}{-}{RRR} \\ % .notdef.  
3544 \testkern{LLL}{-}{210}{-}{RRR} \\ % .notdef.  
3545 \testkern{LLL}{-}{211}{-}{RRR} \\ % .notdef.  
3546 \testkern{LLL}{-}{212}{-}{RRR} \\ % .notdef.  
3547 \testkern{LLL}{-}{213}{-}{RRR} \\ % .notdef.  
3548 \testkern{LLL}{-}{214}{-}{RRR} \\ % .notdef.  
3549 \testkern{LLL}{-}{215}{-}{RRR} \\ % .notdef.  
3550 \testkern{LLL}{-}{216}{-}{RRR} \\ % .notdef.  
3551 \testkern{LLL}{-}{217}{-}{RRR} \\ % .notdef.  
3552 \testkern{LLL}{-}{218}{-}{RRR} \\ % .notdef.  
3553 \testkern{LLL}{-}{219}{-}{RRR} \\ % .notdef.  
3554 \testkern{LLL}{-}{220}{-}{RRR} \\ % .notdef.  
3555 \testkern{LLL}{-}{221}{-}{RRR} \\ % .notdef.  
3556 \testkern{LLL}{-}{222}{-}{RRR} \\ % .notdef.  
3557 \testkern{LLL}{-}{223}{-}{RRR} \\ % .notdef.  
3558 \testkern{LLL}{-}{224}{-}{RRR} \\ % .notdef.  
3559 \testkern{LLL}{-}{225}{-}{RRR} \\ % .notdef.  
3560 \testkern{LLL}{-}{226}{-}{RRR} \\ % .notdef.  
3561 \testkern{LLL}{-}{227}{-}{RRR} \\ % .notdef.

```

3562 \testkern{LLL}{-}{228}{-}{RRR} \ \ % .notdef .
3563 \testkern{LLL}{-}{229}{-}{RRR} \ \ % .notdef .
3564 \testkern{LLL}{-}{230}{-}{RRR} \ \ % .notdef .
3565 \testkern{LLL}{-}{231}{-}{RRR} \ \ % .notdef .
3566 \testkern{LLL}{-}{232}{-}{RRR} \ \ % .notdef .
3567 \testkern{LLL}{-}{233}{-}{RRR} \ \ % .notdef .
3568 \testkern{LLL}{-}{234}{-}{RRR} \ \ % .notdef .
3569 \testkern{LLL}{-}{235}{-}{RRR} \ \ % .notdef .
3570 \testkern{LLL}{-}{236}{-}{RRR} \ \ % .notdef .
3571 \testkern{LLL}{-}{237}{-}{RRR} \ \ % .notdef .
3572 \testkern{LLL}{-}{238}{-}{RRR} \ \ % .notdef .
3573 \testkern{LLL}{-}{239}{-}{RRR} \ \ % .notdef .
3574 \testkern{LLL}{-}{240}{-}{RRR} \ \ % .notdef .
3575 \testkern{LLL}{-}{241}{-}{RRR} \ \ % .notdef .
3576 \testkern{LLL}{-}{242}{-}{RRR} \ \ % .notdef .
3577 \testkern{LLL}{-}{243}{-}{RRR} \ \ % .notdef .
3578 \testkern{LLL}{-}{244}{-}{RRR} \ \ % .notdef .
3579 \testkern{LLL}{-}{245}{-}{RRR} \ \ % .notdef .
3580 \testkern{LLL}{-}{246}{-}{RRR} \ \ % .notdef .
3581 \testkern{LLL}{-}{247}{-}{RRR} \ \ % .notdef .
3582 \testkern{LLL}{-}{248}{-}{RRR} \ \ % .notdef .
3583 \testkern{LLL}{-}{249}{-}{RRR} \ \ % .notdef .
3584 \testkern{LLL}{-}{250}{-}{RRR} \ \ % .notdef .
3585 \testkern{LLL}{-}{251}{-}{RRR} \ \ % .notdef .
3586 \testkern{LLL}{-}{252}{-}{RRR} \ \ % .notdef .
3587 \testkern{LLL}{-}{253}{-}{RRR} \ \ % .notdef .
3588 \testkern{LLL}{-}{254}{-}{RRR} \ \ % .notdef .
3589 \testkern{LLL}{-}{255}{-}{RRR} \ \ % .notdef .
3590 \fi
3591 \end{kerntable}
3592
3593 \end{document}
3594 </template & ot1>

```

That's it.

## Change History

|      |  |   |    |
|------|--|---|----|
| 1.00 |  | by PostScript name . . . . .  | 1  |
|      | General: Total new implementation                                | Find and complain inconsistent kerning data . . . . .                                 | 1  |
| 1.10 |  | Introduce glyph classes to set the kerning for similar glyph shapes by once . . . . . | 1  |
|      | General: Add option ‘writeall’ . . .                             | Reduce left margin . . . . .  | 1  |
|      | Load configuration file if available . . . . .                   | Write glyph name to table . . . .   | 1  |
|      | \writemtxkern: Handle optional comment . . . . .                 |   | 32 |
| 1.11 |  | 1.30  |    |
|      | General: Added more literature to bibliography . . . . .         | General: Add encodings OT1, T2A, T2B, and LY1 . . . . .                               | 1  |
| 1.20 |  | Parameter-dependent encodings (for example with or without ligatures) . . . . .       | 1  |
|      | General: Allow to give glyphs by number (decimal, hex, octal) or |   |    |

|   |    |   |    |
|---|----|---|----|
| Template for OT1 encoding . . .   | 1  | Direct access on glyph names instead of parsing <code>\getpsname</code> .         | 20 |
| 1.31  |    | Do not scale Helvetica by default because this breaks testing Helvetica . . . . . | 17 |
| General: Use <code>\setleftkerning</code> and <code>\setrightkerning</code> instead of writing every kerning pair . . . . | 1  | <code>\defglyphclass:</code> Speed up   |    |
| 1.32  |    | <code>\defglyphclass</code> . . . . .   | 34 |
| <code>\@ifglyphinclass:</code> Speed up   |    | <code>\ifglyphinclass:</code> Speed up  |    |
| <code>\@ifglyphinclass</code> . . . . .   | 37 | <code>\ifglyphinclass</code> . . . . .  | 37 |
| General: Allow to set relative kerning widths in glyph classes . . . .  | 1  | <code>\writemtxkern:</code> Speed up  |    |
|   |    | <code>\writemtxkern</code> . . . . .  | 32 |

## Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

|   |   |  |
|---|---|--|
| <b>Symbols</b>                            | <code>\@oddfont</code> 22, 25, 97, 98                           | 770, 908, 910, 919                                   |
| <code>\@@parseglyphname</code> . . . . .  | 855, 862  | <code>\@tmpglyph</code> 686, 688, 702, 704, 912, 917 |
| <code>\@current</code> . . . . .          | 45  | <code>\@tmpglyph@scaling</code> . . . . .            |
| <code>\@currname</code> . . . . .         | 45  | 697, 699, 917  |
| <code>\@evenfont</code> . . . . .         | 25, 98  | <code>\@unprocessedoptions</code> . . . . .          |
| <code>\@firstglyphname</code> . . . . .   | 615, 619, 654   | 110  |
| <code>\@firstglyphname@</code> . . . . .  | 658   | <code>\@writeclass</code> . . . . .                  |
| <code>\@forallinclass</code> . . . . .    | 634, 635, 685, 833, 835, 910                                    | 905, 925, 927  |
| <code>\@forallinclasses</code> . . . . .  | <u>835</u>  | <code>\@</code> . . . . .                            |
| <code>\@getclassname</code> . . . . .     | 624, 631, 708, 779, 908, 919                                    | 595, 606, 2726–2981, 3013–3268, 3333–3462, 3464–3589 |
| <code>\@gobble</code> . . . . .           | 114   | <b>A</b>   |
| <code>\@ifglyphinclass</code> . . . . .   | 702, 806, <u>809</u>  | <code>\addtolength</code> . . . . .                  |
| <code>\@ifundefined</code> . . . . .      | 665, 789, 811   | <code>\and</code> . . . . .                          |
| <code>\@kernlena</code> . . . . .         | 318, 352, 358, 431, 446, 457, 472, 569                          | <code>\ast</code> . . . . .                          |
| <code>\@kernlenb</code> . . . . .         | 319, 394, 400, 437, 451, 463, 519, 574                          | <code>\AtBeginDocument</code> . . . . .              |
| <code>\@minussign</code> . . . . .        | 313, 352, 394, 431, 437, 446, 451, 457, 463, 472, 519, 569, 574 | 134, 664, 736, 751, 767, 892                         |
| <code>\@nextclass</code> 874, 880, 887    |   | <code>\AtEndDocument</code> . . . . .                |
| <code>\@nextglyphinclass</code> . . . . . | 845, 851, <u>855</u> , <u>887</u>                               | 899  |
|   |   | <code>\AtEndOfClass</code> . . . . .                 |
|   |   | 103, 120   |
|   |   | <b>B</b>   |
|   |   | <code>baselineskip</code> (class option) . . . . .   |
|   |   | <u>6</u>   |
|   |   | <code>\begin</code> . . . . .                        |
|   |   | 585, 2714, 2725, 3006, 3012, 3326, 3332              |
|   |   | <code>\boolean</code> . . . . .                      |
|   |   | 495, 542   |
|   |   | <b>C</b>   |
|   |   | <code>\c@@glyphslot</code> . . . . .                 |
|   |   | 147, 149, 150,                                       |

|  |  |  |  |
|--|--|--|--|
| 291, 297, 301,<br>304–306, 311,<br>615, 617, 687, 694  | <code>\c@slota</code> . . . . . 321,<br>342, 348, 353,<br>355, 358, 361,<br>365, 367, 428,<br>430, 445, 456,<br>470, 488, 490,<br>496, 513, 564, 568 | <code>\ClassError</code> 113, 122,<br>322, 329, 336,<br>364, 406, 688,<br>704, 740, 761,<br>790, 795, 812, 819 | 620, 627, 672,<br>716, 739, 754,<br>770, 799, 823,<br>846, 848, 875, 877   |
| <code>\c@slotb</code> . . . . . 328,<br>343, 349, 354,<br>356, 359, 361,<br>365, 368, 383,<br>389, 395, 397,<br>400, 403, 407,<br>409, 426, 428,<br>436, 445, 462,<br>470, 489, 491,<br>497, 513, 517,<br>535, 537, 543,<br>560, 564, 566, 573 | <code>\ClassWarning</code> . 360, 402  | <code>\ClassWarningNoLine</code><br>. 89, 140, 164, 172  | <code>example</code> (class option) . 7  |
| <code>\c@slotc</code> . . . . . 335,<br>384, 390, 396,<br>398, 401, 403,<br>407, 410, 428,<br>442, 445, 468,<br>517, 536, 538,<br>544, 560, 566, 578   | <code>\closeout</code> . . . . . 933   | <code>\color</code> . . . . . 444,<br>447, 449, 452, 454   | <code>extraname</code> (class option) . . . . . 7  |
| <code>\c@tempcnta</code> . . . . .<br>. 794, 800, 818, 824   | <code>color</code> (class option) . . 7  | <code>copyquotation</code> (class option) . . . . . 7  | <b>F</b>   |
| <code>\char</code> . 311, 428, 430,<br>436, 442, 445,<br>456, 462, 468,<br>470, 517, 564,<br>566, 568, 573, 578  | <code>\CurrentOption</code> . . . . . 49   | <code>\DeclareRobustCommand</code><br>. . . . . 295  | <code>\f@encoding</code> 160, 194, 214   |
| class options:   | <b>D</b>   | <code>\defglyphclass</code> . . 9,<br>661, 746, 759, 775   | <code>\f@family</code> . 160, 194, 216   |
| <code>baselineskip</code> . . . . . 6  | <code>\dagger</code> . . . 24, 510, 557  | <code>\define@key</code> . . . . .<br>. . . 51–55, 59–<br>64, 73, 87, 94,<br>121, 939, 1501, 1538              | <code>\f@series</code> . 161, 194, 218   |
| <code>color</code> . . . . . 7   | <code>\day</code> . . . . . 205  | <code>\definecolor</code> . . . 36,<br>37, 66, 67, 69, 70  | <code>\f@shape</code> . . 161, 194, 220  |
| <code>copyquotation</code> . . . 7   | <code>\DeclareRobustCommand</code><br>. . . . . 295  | <code>designsize</code> (class option) . . . . . 6   | <code>\f@size</code> 161, 162, 170, 176  |
| <code>designsize</code> . . . . . 6  | <code>\defglyphclass</code> . . 9,<br>661, 746, 759, 775   | <code>\documentclass</code> . . . . .<br>. . . . . 123, 126,<br>2600, 2990, 3277                               | <code>family</code> (class option) . 5   |
| <code>encoding</code> . . . . . 5  | <code>\define@key</code> . . . . .<br>. . . 51–55, 59–<br>64, 73, 87, 94,<br>121, 939, 1501, 1538  | <b>E</b>   | <code>\familydefault</code> . . . . . 13   |
| <code>example</code> . . . . . 7   | <code>\definecolor</code> . . . 36,<br>37, 66, 67, 69, 70  | <code>encoding</code> (class option) 5   | <code>\first</code> 639, 645, 646, 648   |
| <code>extraname</code> . . . . . 7   | <code>designsize</code> (class option) . . . . . 6   | <code>\encodingsetup</code> . . . . .<br>. . . 8, 891, 898,<br>2611, 3288, 3290                                | <code>\first@scaling</code> . . . . 644  |
| <code>family</code> . . . . . 5  | <code>\documentclass</code> . . . . .<br>. . . . . 123, 126,<br>2600, 2990, 3277   | <code>\end</code> . . . . . 609, 2982,<br>2984, 3269,<br>3271, 3591, 3593                                      | <code>\firstglyphinclass</code> .<br>. . . . . 719, 912, 917   |
| <code>footer</code> . . . . . 7  | <code>\endfoot</code> . . . . . 607  | <code>\endfoot</code> . . . . . 607  | <code>\fontsize</code> . . . . . 155, 584  |
| <code>italicizing</code> . . . . . 9   | <code>\endhead</code> . . . . . 596  | <code>\endmetrics</code> . . . . . 929   | <code>footer</code> (class option) . 7   |
| <code>ligaturing</code> . . . . . 8, 9   | <code>\endmetrics</code> . . . . . 929   | environments:  | <code>\forallclasses</code> . . . . .<br>. . . 671, 701, 738,<br>753, 769, 864, 906  |
| <code>papersize</code> . . . . . 7   | environments:  | <code>kerntable</code> . . . . . 582   | <code>\forallinclass</code> . . . . 832  |
| <code>series</code> . . . . . 5  | <code>kerntable</code> . . . . . 3   | <code>\equal</code> . . . . . 447, 452,<br>473, 495, 509,<br>512, 520, 542,<br>556, 559, 618,                  | <code>\forallinclasses</code> . . 832  |
| <code>shape</code> . . . . . 6   | <code>\equal</code> . . . . . 447, 452,<br>473, 495, 509,<br>512, 520, 542,<br>556, 559, 618,  |  | <b>G</b>   |
| <code>size</code> . . . . . 6  |  |  | <code>\gdef</code> . . . . . 193,<br>344, 346, 385, 387  |
| <code>writeall</code> . . . . . 7  |  |  | <code>\geometry</code> . . . . . 62  |
|  |  |  | <code>\getclassofglyph</code> 619,<br>626, 780, 800, 824   |
|  |  |  | <code>\getkern</code> . . 266, 470, 517  |
|  |  |  | <code>\getpsname</code> . . . 144,<br>149, 342, 343,<br>348, 349, 353–<br>356, 358, 359,<br>361, 365, 367,<br>368, 383, 384,<br>389, 390, 395–<br>398, 400, 401,<br>403, 407, 409,<br>410, 426, 488–<br>491, 535–538,<br>615, 617, 694,<br>800, 824, 978,<br>1239, 1553,<br>1814, 2075, 2336 |
|  |  |  | <code>\getpsunit</code> . . . . . 263  |
|  |  |  | <code>\getslotnumber</code> 299, 424   |
|  |  |  | <code>\global</code> 157, 197, 198, 696  |

|   |   |  |  |
|---|---|--|--|
|   | <b>H</b>  | 2994, 2997,  | <code>\makebox</code> . . . . . 445,   |
| <code>\hfill</code> . . . . . 24  |   | 3280, 3281, 3284   | 492, 510, 539, 557   |
|   | <b>I</b>  | <code>kerntable</code> (environ-<br>ment) . . . . . 3, <u>582</u>                                    | <code>\markboth</code> . . . . . 196, 197  |
| <code>\if@glyphfound</code> . . . . 787   |   | <code>\krntst@round</code> 254, 256  | <code>\markright</code> . . . . . 198  |
| <code>\if@tempswa</code> 676, 715,  |   | <code>\krntst@baselineskip</code><br>. . . . . 28,   | <code>\metrics</code> . . . . . 225  |
| 745, 758, 774, 911  |   | 33, 57, 59, 155, 584   | <code>\month</code> . . . . . 205  |
| <code>\if@tempswb</code> 612, 636, 695  |   | <code>\krntst@designsize</code> .<br>. . . . . 39, 60, 156, 167                                      | <code>\mtxcomment</code> . . . . . 8, <u>129</u>   |
| <code>\ifcase</code> . . . . .  |   | <code>\krntst@encoding</code> . .<br>. . . . . 29, 51, 135,  | <code>\mtxfile</code> 130, 200–202,<br>204, 207, 208,<br>210, 211, 213,<br>215, 217, 219,<br>221, 223–228,<br>230, 232, 233,<br>235, 237, 240,<br>637, 652, 901,<br>902, 904, 915,<br>926, 928–931, 933  |
| 979, 1240, 1554,  |   | 137, 154, 158, 583   | <code>\mtxfilename</code> 192, 193,<br>195, 196, 199,<br>200, 203, 932, 934  |
| 1815, 2076, 2337  |   | <code>\krntst@example</code> . . .<br>. . . . . 34, 61, 565, 573                                     |  |
| <code>\ifdim</code> . . . . 162, 170,   |   | <code>\krntst@extraname</code> .<br>. . . . . 35, 63, 194, 222                                       |  |
| 475, 476, 482,  |   | <code>\krntst@family</code> . 52,<br>112, 154, 158, 583  |  |
| 483, 497, 500,  |   | <code>\krntst@helpfont</code> . .<br>. . . . . 132, 424, 426,  | <b>N</b>   |
| 503, 504, 522,  |   | 471, 518, 580,   | <code>\needsfontinstversion</code><br>. . . . . 226  |
| 523, 529, 530,  |   | 586–595, 597–606   | <code>\newboolean</code> . . . . . 38  |
| 544, 547, 550, 551  |   | <code>\krntst@round</code> . 241, 260  | <code>\newglyphclass</code> 9, 75–<br>84, <u>735</u> , 2622,<br>2623, 2625,<br>2627, 2628,<br>2630, 2632,<br>2633, 2635,<br>2637, 2640,<br>2641, 2643,<br>2648, 2651,<br>2653, 2656,<br>2658, 2659,<br>2661, 2662,<br>2664, 2666,<br>2669, 2670,<br>2672, 2673,<br>2675–2678,<br>2680, 2681,<br>2683, 2685–<br>2688, 2690,<br>2691, 2693,<br>2694, 2696,<br>2697, 2699–<br>2702, 2704,<br>2705, 2707,<br>2708, 2711,<br>3301, 3303,<br>3305, 3307,<br>3308, 3310,<br>3311, 3313, |
| <code>\iffalse</code> . . . . . 3463  |   | <code>\krntst@series</code> . . . . .<br>30, 53, 154, 159, 583                                       |  |
| <code>\ifglyphinclass</code> . . . . <u>788</u>   |   | <code>\krntst@shape</code> . . . . .<br>31, 54, 154, 159, 583  |  |
| <code>\ifthenelse</code> . . . . .  |   | <code>\krntst@side</code> . . . . .<br>. . . . . 814, 815, 824                                       |  |
| . 447, 452, 473,  |   | <code>\krntst@size</code> . . . . 27,<br>32, 33, 56, 57,<br>155, 159, 162, 584                       |  |
| 495, 509, 512,  |   | <code>\kt@tempa</code> . . . . .<br>. . . . . 841, 843, 845,<br>850, 851, 870,<br>872, 874, 879, 880 |  |
| 520, 542, 556,  |   | <code>\KVo@tempa</code> . . . . 42, 43, 48   |  |
| 559, 618, 620,  |   |  |  |
| 627, 672, 716,  |   |  |  |
| 739, 754, 770,  |   |  |  |
| 799, 823, 848, 877  |   |  |  |
| <code>\ignorespaces</code> . . . . . 580  |   |  |  |
| <code>\ignorespacesafterend</code><br>. . . . . 610   |   |  |  |
| <code>\immediate</code> . . . . . 200   |   |  |  |
| <code>\input</code> . 2618, 3004, 3297  |   |  |  |
| <code>\InputIfFileExists</code> .<br>. . . . . 104, 139   |   |  |  |
| <code>italicizing</code> (class op-<br>tion) . . . . . 9  |   |  |  |
|   | <b>J</b>  |  |  |
| <code>\jobname</code> . . . . . 212   |   |  |  |
|   | <b>K</b>  |  |  |
| <code>\kern</code> . . . . . 268, 347,<br>388, 434, 440,<br>460, 466, 571, 576  |   |  |  |
| <code>\kernlen</code> . . . . 256, 475–<br>477, 479, 482–<br>484, 495, 497,<br>500, 503–505,<br>522–524, 526,<br>529–531, 542,<br>544, 547, 550–552 |   |  |  |
| <code>\kernsetup</code> . . . . . 5,<br><u>101</u> , 2603, 2604,<br>2607, 2993,   |   |  |  |
|   | <b>L</b>  |  |  |
|   | <code>\large</code> 486, 507, 533, 554                          |  |  |
|   | <code>\leftkern</code> . . . . 626,<br>627, 629, 631, 635       |  |  |
|   | <code>ligaturing</code> (class op-<br>tion) . . . . . 8, 9      |  |  |
|   | <code>\listfiles</code> . . . . .<br>. . . . . 2597, 2987, 3274 |  |  |
|   | <code>\LoadClass</code> . . . . . 11                            |  |  |
|   | <code>\lowercase</code> . . . . 137, 193                        |  |  |
|   | <b>M</b>  |  |  |
|   | <code>\makeatletter</code> 938, 1500                            |  |  |
|   | <code>\makeatother</code> . 977, 1552                           |  |  |

|   |  |   |
|---|--|---|
| 3314, 3316,<br>3317, 3319,<br>3321, 3322, 3324  | \provideglyphclass .<br>..... 9, <u>766</u>                                      | \strip@pt .... 253, 259   |
| \newwrite ..... 240   | \ProvidesClass ..... 1   | <b>T</b>  |
| \noexpand . 44, 137, 193  | \ProvidesFile ..... 2–8  | \testkern .....<br>. 3, <u>317</u> , 2726–<br>2981, 3013–<br>3268, 3333–<br>3462, 3464–3589   |
| \normalfont ..... 132   | \psunit 156, 157, 169,<br>262, 264, 350,<br>391, 434, 440,<br>460, 466, 571, 576 | \textbf ..... 302   |
| \normalsize ..... 132   |  | \textcolor .....<br>. 474, 481, 486,<br>502, 507, 521,<br>528, 533, 549, 554  |
| \not .. 495, 542, 846, 875  |  | \textleft . 628, 631, 640   |
| \number ..... 205   |  | \textright 621, 624, 640  |
|   | <b>R</b>   | \the@firstscale ... 699   |
| <b>O</b>  | \ratio ..... 264, 269  | \the@tmpscale .....<br>. 645, 647, 718, 723   |
| \oldkerna .... 344,<br>346, 432, 447,<br>458, 473, 509, 512                           | \renewglyphclass 9, <u>750</u>   | \thisglyphname ....<br>. 692, 694, 705,<br>718, 719, 723, 726   |
| \oldkernb .... 385,<br>387, 438, 452,<br>464, 520, 556, 559                           | \RequirePackage ...<br>... 12, 14–20, 135  | \typeout ..... 158,<br>160, 168, 199,<br>353, 395, 645,<br>731, 858, 893,<br>900, 907, 934,<br>940, 972–974,<br>976, 1502, 1539,<br>1546–1549, 1551 |
| \openout ..... 200  | \reserved@f ... 136–140  |   |
| \or . 980–1234, 1241–<br>1495, 1555–<br>1809, 1816–<br>2070, 2077–<br>2331, 2338–2592 | \rightkern ... 619,<br>620, 622, 624, 634  |   |
|   | \rnd@tempa .....<br>. 253, 254, 259, 260   |   |
|   | \round ..... <u>257</u> , 270  |   |
|   | <b>S</b>   |   |
|   | \saveslotnumber ...<br>. <u>295</u> , 320, 327,<br>334, 686, 793, 817            |   |
|   | \second 639, 645, 646, 648   |   |
|   | \second@scaling ... 644  |   |
|   | \selectfont ... 155, 584   |   |
|   | series (class option) . 5  |   |
|   | \set ..... 916   |   |
|   | \setboolean ..... 88   |   |
|   | \setkern .. 234, 638, 653  |   |
|   | \setkeys ..... 44,<br>101, 895, 975, 1550  |   |
|   | \sfdefault ..... 13  |   |
|   | shape (class option) .. 6  |   |
|   | size (class option) ... 6  |   |
|   | \small .... 474, 481,<br>502, 521, 528, 549                                      |   |
|   | \stop ..... 118  |   |
|   | \stoploop . 836, 865, <u>884</u>   |   |
|   |  | <b>U</b>  |
|   |  | \upshape ..... 196  |
|   |  | \usefont ..... 154, 583   |
|   |  | <b>W</b>  |
|   |  | \whiledo .. 147, 846, 875   |
|   |  | writeall (class option) 7   |
|   |  | \writetxkern . 496,<br>513, 543, 560, <u>612</u>  |
|   |  | <b>Y</b>  |
|   |  | \year ..... 205   |