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 LaTeX. 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 LaTeX.

Contents

Copyright

1 Introduction

Every glyph of a font is surrounded by a bouding 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 kernig pairs are missing. Have a look at this example with quotation marks used in German, with the original kerning of the Times:

„TA“ »VaV«

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):

A“ A“ A“ A“ A“ A“ A“ A“ A“ A“

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` `\testkern`

The kerning table is generated by the `kerntable` environment which is represented by a `longtable` environment internally. Each line of this environment has 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

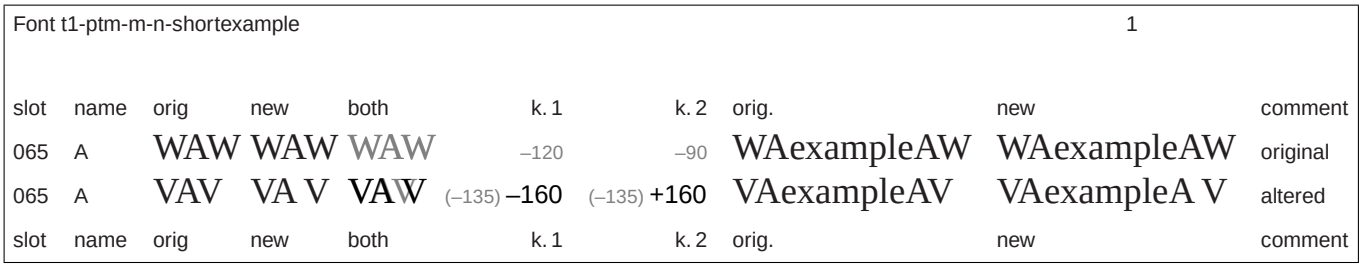


Figure 1: Example for a part of a kerning table (explanations in the text)

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 LaTeX 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 Type 1 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 forth 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`[1] 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.

[1] 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.

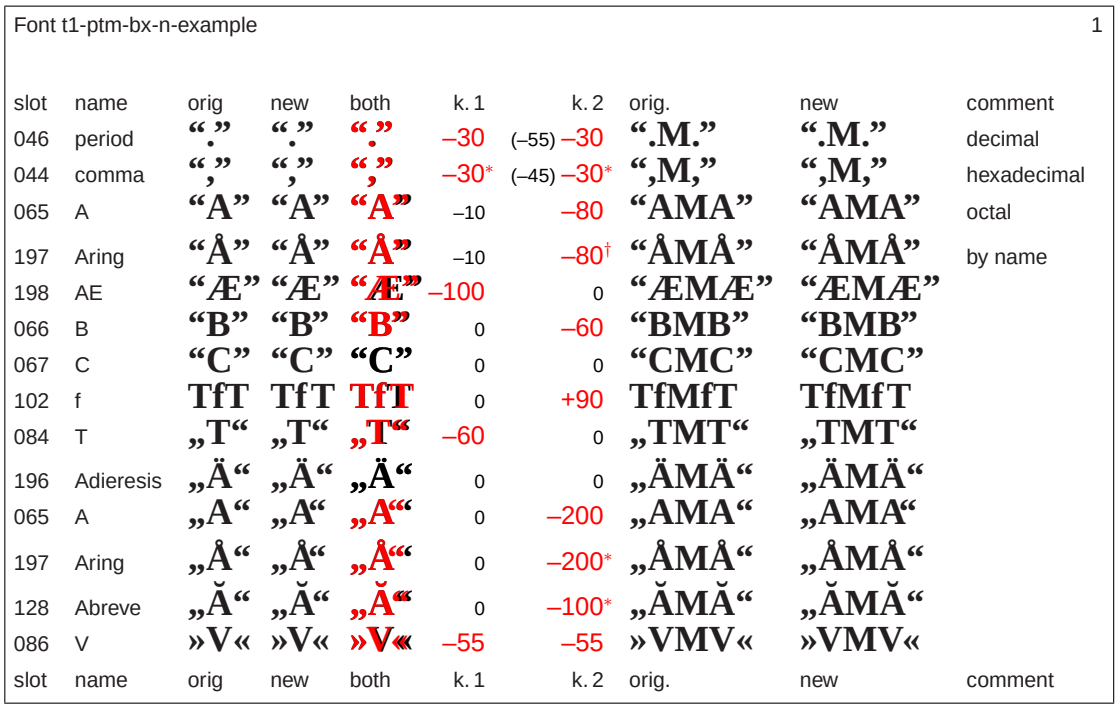


Figure 2: Example for a part of a kerning table (explanations in the text). The labels marked with a star are described in Section 2.4.

`shape` `shape=`⟨*font shape*⟩**:** Abbreviation for the font shape (default: `n`), e.g., `n` for upright, `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 LaTeX. 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.[2]

`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.[3]

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.
%%
```

[2] This is why the first example used this option.

[3] 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`, `ffl` 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:

`ligaturing` `ligaturing=`⟨*number*⟩**:** Level of how many ligatures are used (`0`, `1`, or `2`, default `2`). Please refer to `ot1.etx` for more details.

`italicizing` `italicizing=`⟨`true`/`false`⟩**:** Use `dollar` when `false` and `sterling` when `true` (default: `false`).

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, Ď, Đ, E, Ě, Ę, È, É, Ê, Ë, F, H, I, İ, Ì, Í, Î, Ï, IJ, J, K, L, Ĺ, Ľ, Ł, N, Ń, Ň, Ñ, P, R, Ŕ, Ř, Ŋ, 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:[4]

```
%\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

[4] 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 occurance of this glyph, the correct kerning data are not shown for the occurances 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 "Å", but not for "Ä". 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 kerningn width".

For example,

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

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

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

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 LaTeX 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”, “Tå”, “Tą” 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”.[5]) 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}{-}{`⟨*glyph*⟩`}{-}{020}`, `\testkern{020}{-}{`⟨*glyph*⟩`}{-}{019}`, `\testkern{016}{-}{`⟨*glyph*⟩`}{-}{017}`, `\testkern{018}{-}{`⟨*glyph*⟩`}{-}{016}`, `\testkern{017}{-}{`⟨*glyph*⟩`}{-}{018}`, where ⟨*glyph*⟩ 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

[5] 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.[6]

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 `c059013l.afm` and `c059013l.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
\needsfontinstversion{1.914}
% input AFMs:
\transformfont{9ncr8r}{\reencodefont{8r}{\fromafm{c059013l}}}
\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
```

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

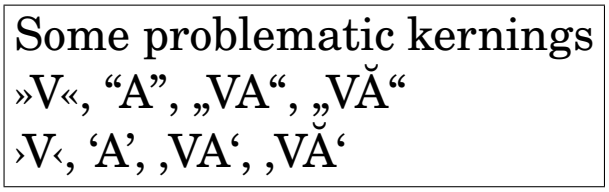


Figure 3: Font example for Century Schoolbook L with original kerning

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

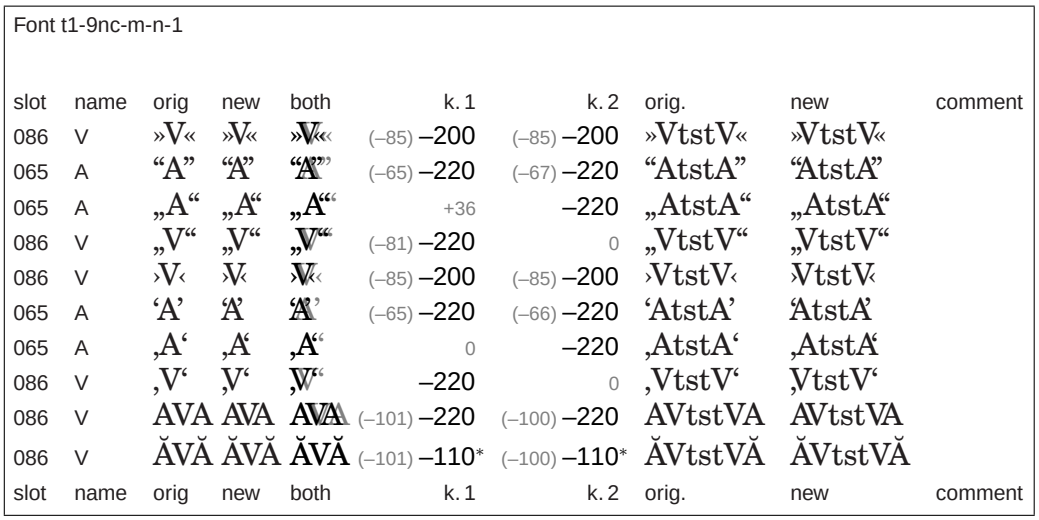
```
9ncr8r CenturySchL-Roma "TeXBase1Encoding ReEncodeFont" <8r.enc <c059013l.pfb
```

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 pdfLaTeX 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‘, Ă“, and ‚V.

Emagine 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	ĂVĂ	ĂVĂ	ĂVĂ	(–101) –110*	(–100) –110*	ĂVtstVĂ	ĂVtstVĂ	

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{c059013l}}}
\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

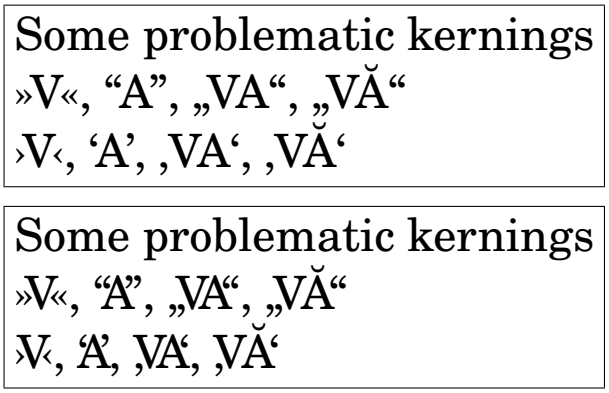


Figure 5: Font example for Century Schoolbook L with original (top) and modified (bottom) kerning. The kerning is much too strong. Here, it only shows the effect of altering the kerning.

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 Ă 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}
```

```
⟨mtx & t2a⟩\ProvidesFile{t2amtx.clo}
⟨mtx & t2b⟩\ProvidesFile{t2bmtx.clo}
⟨mtx & ly1⟩\ProvidesFile{ly1mtx.clo}
⟨version⟩\ProvidesFile{krntst-v.tex}
⟨class | mtx | version⟩  [2004/04/14  v1.32  Generate kerning tables]
```

6.1 Class file

Use a standard class as base.

```
⟨*class⟩
\LoadClass[10pt]{article}
```

Use most of the space on the paper.

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

Font for the legends.

```
\renewcommand*\familydefault{\sfdefault}
\RequirePackage{helvet}
```

More required packages.

```
\RequirePackage{calc}
\RequirePackage{longtable}
\RequirePackage{array}
\RequirePackage{color}
\RequirePackage{ifthen}
\RequirePackage{keyval}
```

Layout settings.

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

Declare lengths for the font size and the baselineskip.

```
\newlength\krntst@size
\newlength\krntst@baselineskip
```

Set the default values for the class options.

```
\def\krntst@encoding{T1}
\def\krntst@series{m}
\def\krntst@shape{n}
\setlength\krntst@size{17.28pt}
\setlength\krntst@baselineskip{1.2\krntst@size}
\def\krntst@example{example}
\def\krntst@extraname{}
\definecolor{oldcolor}{gray}{0.5}
\definecolor{newcolor}{gray}{0}
\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.

```
\newcommand\krntst@designsize{1em}
```

Process the class options using the `keyval` package.

```
\def\ProcessOptionsWithKV#1{%
  \let\@tempc\relax
  \let\KVo@tempa\@empty
  \edef\KVo@tempa{%
    \noexpand\setkeys{#1}{%
      \@ptionlist{\@currname.\@currext}%
    }%
  }%
  \KVo@tempa
  \let\CurrentOption\@empty
}
```

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

```
\define@key{krntst}{encoding}{\def\krntst@encoding{#1}}
\define@key{krntst}{family}{\def\krntst@family{#1}}
\define@key{krntst}{series}{\def\krntst@series{#1}}
\define@key{krntst}{shape}{\def\krntst@shape{#1}}
\define@key{krntst}{size}{%
  \setlength\krntst@size{#1}%
  \setlength\krntst@baselineskip{1.2\krntst@size}%
}
\define@key{krntst}{baselineskip}{\setlength\krntst@baselineskip{#1}}
\define@key{krntst}{designsize}{\def\krntst@designsize{#1}}%
\define@key{krntst}{example}{\def\krntst@example{#1}}
\define@key{krntst}{papersize}{\geometry{#1}}
\define@key{krntst}{extraname}{\def\krntst@extraname{-#1}}
\define@key{krntst}{color}[true]{%
  \csname if#1\endcsname
    \definecolor{oldcolor}{gray}{0}%
    \definecolor{newcolor}{rgb}{1,0,0}%
  \else
    \definecolor{oldcolor}{gray}{0.5}%
    \definecolor{newcolor}{gray}{0}%
  \fi
}
```

Do the copying of quotation marks by introducing glyph classes.

```
\define@key{krntst}{copyquotation}[true]{%
  \csname if#1\endcsname
    \newglyphclass{left}{leftguillemots}{guillemotleft,guilsinglleft}%
    \newglyphclass{right}{leftguillemots}{guillemotleft,guilsinglleft}%
    \newglyphclass{left}{rightguillemots}{guillemotright,guilsinglright}%
    \newglyphclass{right}{rightguillemots}{guillemotright,guilsinglright}%
    \newglyphclass{left}{leftquotes}{quotedblleft,quoteleft}%
    \newglyphclass{right}{leftquotes}{quotedblleft,quoteleft}%
    \newglyphclass{left}{rightquotes}{quotedblright,quoteright}%
    \newglyphclass{right}{rightquotes}{quotedblright,quoteright}%
    \newglyphclass{left}{basequotes}{quotedblbase,quotesinglbase}%
    \newglyphclass{right}{basequotes}{quotedblbase,quotesinglbase}%
  \fi
}
\define@key{krntst}{writeall}[true]{%
  \setboolean{krntst@writeall}{#1}%
  \ClassWarningNoLine{kerntest}{You are writing the new and the
```

```
    original kerning data\MessageBreak
    to the mtx file (option 'writeall'). Normally, it is\MessageBreak
    not necessary to write original data}%
}
\define@key{krntst}{footer}[true]{%
  \csname if#1\endcsname
  \else
    \def\@oddfoot{}%
    \def\@evenfoot{\@oddfoot}%
  \fi
}
```

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

```
\newcommand\kernsetup{\setkeys{krntst}}
\@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.

```
\AtEndOfClass{%
  \InputIfFileExists{kerntest.cfg}{%
    \message{Configuration file 'kerntest.cfg' loaded.}%
  }{%
    \message{No configuration file 'kerntest.cfg' found.}%
  }
```

Now, process the class options.

```
  \ProcessOptionsWithKV{krntst}
```

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

```
  \let\@unprocessedoptions\relax
}
```

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

```
\ifx\krntst@family\relax
  \ClassError{kerntest}{Class option family not or incorrect
    given\@gobble}{%
    You have to specify the font family by using the
    class\MessageBreak
    option family=<fontfamily>}%
  \stop
\fi
```

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

```
\AtEndOfClass{%
  \define@key{krntst}{family}{%
    \ClassError{kerntest}{Option 'family' used outside
      \string\documentclass\space command}{%
      The option 'family=<fontfamily>' has to be specified in the
      optional argument\MessageBreak
      of the \string\documentclass\space command.}%
  }
}
```

\mtxcomment Define a command that writes a comment to the `mtx` file.

```
\newcommand\mtxcomment[1]{%
  \protected@write\mtxfile{}{\@percentchar\space #1}%
}
```

Define a command that is used to access the font for the legends.

```
\newcommand\krntst@helpfont{\normalfont\normalsize}
```

An internal counter that stores the slot of a glyph.

```
\newcounter{@glyphslot}%
```

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

```
\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.

```
  \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.

```
  \edef\reserved@f{%
    \lowercase{\def\noexpand\reserved@f{\krntst@encoding mtx.clo}}}%
  \reserved@f
  \InputIfFileExists\reserved@f{}{%
    \ClassWarningNoLine{kerntest}{Postscript name file `\reserved@f'
      not found.\MessageBreak
      The kerning table will be okay, but the generated mtx file will
      not be usable}%
    \newcommand\getpsname[1]{unknown character `##1'}%
  }%
```

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).

```
  \setcounter{@glyphslot}{0}%
  \whiledo{\the\c@@glyphslot<256}{%
    \expandafter\edef
    \csname slotnumber@glyph@\getpsname{\the\c@@glyphslot}\endcsname{%
      \the\c@@glyphslot}%
    \stepcounter{@glyphslot}%
  }%
```

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

```
  \begingroup
```

Switch to the font that shall be tested to see if the desired font size is possible etc.

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

Set the Postscript font unit to 0.001 of the design size which is 1 em, normally.

```
    \psunit=\krntst@designsize\relax
    \global\psunit=0.001\psunit
```

Give some feedback.

```
    \typeout{Requested: \krntst@encoding-\krntst@family-%
      \krntst@series-\krntst@shape, size \the\krntst@size}%
    \typeout{Using:\space\space\space\space\space \f@encoding-\f@family-%
      \f@series-\f@shape, size \f@size pt}%
    \expandafter\ifdim\the\krntst@size=\f@size pt\relax
    \else
      \ClassWarningNoLine{kerntest}{Using different font size than
        requested}%
    \fi
    \setlength{\@tempdima}{\krntst@designsize}%
    \typeout{Postscript font unit for design size \the\@tempdima:
      \the\psunit}%
    \expandafter\ifdim\the\@tempdima=\f@size pt\relax
    \else
      \ClassWarningNoLine{kerntest}{The design size (\the\@tempdima,
        1em by default,\MessageBreak
        or given value from option 'designsize') of the
        font\MessageBreak
        is not equal to the LaTeX font size (\f@size pt).\MessageBreak
        This can have two reasons:\MessageBreak
        1. The font does not define 1em to be the design
        size\MessageBreak
        \space\space\space (for example, Computer
        Modern).\MessageBreak
        2. The font is implicitely scaled by the fd-file\MessageBreak
        \space\space\space (for example, when using
        helvet.sty).\MessageBreak
        This can cause the PostScript font unit length to
        be\MessageBreak
        incorrect.
        You may set the design size for calculation\MessageBreak
        of the font unit explicitely by using the class\MessageBreak
        option 'designsize'}%
    \fi
```

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

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

Set the page headings.

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

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

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

Open the `mtx` file.

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

Write a nice header to the `mtx` file.

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

```
    \protected@write\mtxfile{}{\@percentchar\@percentchar\space
      This is file `\mtxfilename.mtx',}%
    \protected@write\mtxfile{}{\@percentchar\@percentchar\space
      generated on \number\year/\number\month/\number\day\space
      by kerntest.cls, (c) 2004 Harald Harders.}%
    \protected@write\mtxfile{}{\@percentchar\@percentchar}%
    \protected@write\mtxfile{}{\@percentchar\@percentchar\space
      The original source file was:}%
    \protected@write\mtxfile{}{\@percentchar\@percentchar}%
    \protected@write\mtxfile{}{\@percentchar\@percentchar\space
      \jobname\space (.tex?) with these font options:}%
    \protected@write\mtxfile{}{\@percentchar\@percentchar\space
      Encoding: \f@encoding}%
    \protected@write\mtxfile{}{\@percentchar\@percentchar\space
      Family: \space\space\f@family}%
    \protected@write\mtxfile{}{\@percentchar\@percentchar\space
      Series: \space\space\f@series}%
    \protected@write\mtxfile{}{\@percentchar\@percentchar\space
      Shape: \space\space\space\f@shape}%
    \protected@write\mtxfile{}{\@percentchar\@percentchar\space
      User-defined name: \krntst@extraname}%
    \protected@write\mtxfile{}{\@percentchar\@percentchar}%
    \protected@write\mtxfile{}{\string\relax}%
    \protected@write\mtxfile{}{\string\metrics}%
    \protected@write\mtxfile{}{\string\needsfontinstversion{1.926}}%
    \protected@write\mtxfile{}{\@percentchar\@percentchar}%
    \protected@write\mtxfile{}{\@percentchar\@percentchar\space
      Kerning data for single characters and}%
    \protected@write\mtxfile{}{\@percentchar\@percentchar\space
      the first members of the glyph classes.}%
    \protected@write\mtxfile{}{\@percentchar\@percentchar}%
    \protected@write\mtxfile{}{\@percentchar\@percentchar\space
      After each \string\setkern\space entry, the glyph classes}%
    \protected@write\mtxfile{}{\@percentchar\@percentchar\space
      for both glyphs are given (./. means no class).}%
    \protected@write\mtxfile{}{\@percentchar\@percentchar}%
  \endgroup
}
```

Declare the output handle for the `mtx` file.

```
\newwrite\mtxfile
```

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

```
\def\krntst@round#1.#2#3#4\@empty{%
  \setlength\@tempdimc{#1pt}%
  \if#2.%
  \else
    \ifnum#2>4
      \ifnum#1#2<0
        \addtolength\@tempdimc{-1.1pt}%
      \else
        \addtolength\@tempdimc{1.1pt}%
      \fi
    \fi
```

```
  \fi
  \edef\rnd@tempa{\strip@pt\@tempdimc}%
  \expandafter\krntst@@round\rnd@tempa.000\@empty
}
```

Calculate the rounded length.

```
\def\krntst@@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`.

```
\newcommand*\round[1]{%
  \setlength\@tempdimc{#1}%
  \edef\rnd@tempa{\strip@pt\@tempdimc}%
  \expandafter\krntst@round\rnd@tempa.000\@empty
}
```

Define the Postscript font length.

```
\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.

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

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

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

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

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

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`.

```
\def\@saveslotnumber#1#2\@empty{%
  \if#1"\relax
    \setcounter{@glyphslot}{#1#2}%
  \else
    \if#1'\relax
      \setcounter{@glyphslot}{#1#2}%
    \else
      \ifnum9<1#1\relax
        \setcounter{@glyphslot}{#1#2}%
      \else
        \begingroup\expandafter\expandafter\expandafter\endgroup
```

```
        \expandafter\ifx\csname slotnumber@glyph@#1#2\endcsname\relax
          \setcounter{@glyphslot}{-1}%
        \else
          \setcounter{@glyphslot}{\csname slotnumber@glyph@#1#2\endcsname}%
        \fi
      \fi
    \fi
  \fi
  \ifnum\the\c@@glyphslot>255\relax
    \setcounter{@glyphslot}{-1}%
  \fi
}
```

`\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.

```
\DeclareRobustCommand*\saveslotnumber[2]{%
  \expandafter\@saveslotnumber#2\@empty
  \setcounter{#1}{\the\c@@glyphslot}%
}
```

`\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.

```
\newcommand\getslotnumber[1]{%
  \expandafter\@saveslotnumber#1\@empty
  \ifnum\the\c@@glyphslot<0\relax
    \textbf{???}%
  \else
    \ifnum\c@@glyphslot<100\relax0\fi
    \ifnum\c@@glyphslot<10\relax0\fi
    \the\c@@glyphslot
  \fi
}
```

`\printglyph` Print the glyph with the given PostScript name or slot number (in decimals, octal, or hexadecimals; as usual in LaTeX). 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}
%
```

```
\newcommand*\printglyph[1]{%
  \expandafter\@saveslotnumber#1\@empty
  \char\the\c@@glyphslot
}
```

A help macro for comparing arguments with "-".

```
\edef\@minussign{-}%
```

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

```
\newcounter{@slota}
\newcounter{@slotb}
\newcounter{@slotc}
```

`\testkern` The main macro of the class. It takes 5 arguments:
`{`⟨*glyph 1*⟩`}{`⟨*kerning 1–2*⟩`}{`⟨*glyph 2*⟩`}{`⟨*kerning 2–3*⟩`}{`⟨*glyph 3*⟩`}`.
The glyphs are given by their number, not the glyphs itself.

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

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

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

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

```
  \saveslotnumber{@slota}{#1}%
  \ifnum\the\c@@slota<0%
    \ClassError{kerntest}{Used unknown glyph `#1'}{%
      You may have misspelled the glyph or you have taken an invalid
      number.}%
    \setcounter{@slota}{0}%
  \fi
  \saveslotnumber{@slotb}{#3}%
  \ifnum\the\c@@slotb<0%
    \ClassError{kerntest}{Used unknown glyph `#3'}{%
      You may have misspelled the glyph or you have taken an invalid
      number.}%
    \setcounter{@slotb}{0}%
  \fi
  \saveslotnumber{@slotc}{#5}%
  \ifnum\the\c@@slotc<0%
    \ClassError{kerntest}{Used unknown glyph `#5'}{%
      You may have misspelled the glyph or you have taken an invalid
      number.}%
    \setcounter{@slotc}{0}%
  \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.

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

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

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

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

```
    \gdef\oldkerna{%
      \kern
```

```
      \csname kt@kerning@\getpsname{\the\c@@slota}@%
      \getpsname{\the\c@@slotb}\endcsname
      \psunit
    }%
```

If no new kerning ist given just tell the user that he reuses a kerning.

```
    \ifx\@kernlena\@minussign
      \typeout{Kerning pair for \getpsname{\the\c@@slota}-%
        \getpsname{\the\c@@slotb} reused (value
        \csname kt@kerning@\getpsname{\the\c@@slota}@%
        \getpsname{\the\c@@slotb}\endcsname).}%
    \else
```

Old kerning exists and new kerning, too. Test if the old and new kernings are identical.

```
      \ifnum\@kernlena=\csname kt@kerning@\getpsname{\the\c@@slota}@%
      \getpsname{\the\c@@slotb}\endcsname\relax
```

Yes. Nevertheless, generate a warning.

```
        \ClassWarning{kerntest}{Kerning for
          \getpsname{\the\c@@slota}-\getpsname{\the\c@@slotb}\MessageBreak
          repeated (value #2)}%
      \else
```

No. Produce an erroe message.

```
        \ClassError{kerntest}{Conflicting kerning for
          \getpsname{\the\c@@slota}-\getpsname{\the\c@@slotb}\MessageBreak
          (new value #2 != old value
          \csname kt@kerning@\getpsname{\the\c@@slota}@%
          \getpsname{\the\c@@slotb}\endcsname)%
        }{%
          You have given the kerning for the specified glyph pair
          twice with different\MessageBreak
          values. This may also appear when using glyph classes.
          You have to give the\MessageBreak
          kerning only once per glyph class.\MessageBreak
          You may leave out the second kerning pair, or you may
          give\MessageBreak
          the kerning '-'. Then, the old value is reused.
        }%
      \fi
    \fi
  \fi
```

Second pair.

```
  \begingroup\expandafter\expandafter\expandafter\endgroup
  \expandafter\ifx\csname kt@kerning@\getpsname{\the\c@@slotb}@%
  \getpsname{\the\c@@slotc}\endcsname\relax
    \gdef\oldkernb{}%
  \else
    \gdef\oldkernb{%
      \kern
      \csname kt@kerning@\getpsname{\the\c@@slotb}@%
      \getpsname{\the\c@@slotc}\endcsname
      \psunit
    }%
```

```
%
    \ifx\@kernlenb\@minussign
      \typeout{Kerning pair for \getpsname{\the\c@@slotb}-%
        \getpsname{\the\c@@slotc} reused (value
        \csname kt@kerning@\getpsname{\the\c@@slotb}@%
        \getpsname{\the\c@@slotc}\endcsname).}%
    \else
      \ifnum\@kernlenb=\csname kt@kerning@\getpsname{\the\c@@slotb}@%
      \getpsname{\the\c@@slotc}\endcsname\relax
        \ClassWarning{kerntest}{Kerning for
          \getpsname{\the\c@@slotb}-\getpsname{\the\c@@slotc}\MessageBreak
          repeated (value #4)}%
      \else
        \ClassError{kerntest}{Conflicting kerning for
          \getpsname{\the\c@@slotb}-\getpsname{\the\c@@slotc}\MessageBreak
          (new value #4 != old value
          \csname kt@kerning@\getpsname{\the\c@@slotb}@%
          \getpsname{\the\c@@slotc}\endcsname)%
        }{%
          You have given the kerning for the specified glyph pair
          twice with different\MessageBreak
          values. This may also appear when using glyph classes.
          You have to give the\MessageBreak
          kerning only once per glyph class.\MessageBreak
          You may leave out the second kerning pair, or you may
          give\MessageBreak
          the kerning '-'. Then, the old value is reused.
        }%
      \fi
    \fi
  \fi
```

First, type the slot number of glyph 2.

```
  \krntst@helpfont\getslotnumber{#3}%
  &
```

Type the postscript name of glyph 2.

```
  \krntst@helpfont\getpsname{\the\c@@slotb}%
  &
```

Print the three glyphs with original kerning.

```
  \char\the\c@@slota\char\c@@slotb\char\c@@slotc%
  &
```

Print glyph 1.

```
  \char\the\c@@slota%
```

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

```
  \ifx\@kernlena\@minussign
    \oldkerna
  \else
    \kern#2\psunit
  \fi
```

Print glyph 2.

```
  \char\the\c@@slotb%
```

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

```
\ifx\@kernlenb\@minussign
  \oldkernb
\else
  \kern#4\psunit
\fi
```

Print glyph 3.

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

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

```
\color{oldcolor}%
\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.

```
\ifx\@kernlena\@minussign
  \ifthenelse{\equal{\oldkerna}{}}{}{\color{newcolor}}%
\else
  \color{newcolor}%
\fi
\ifx\@kernlenb\@minussign
  \ifthenelse{\equal{\oldkernb}{}}{}{\color{newcolor}}%
\else
  \color{newcolor}%
\fi
\char\the\c@@slota%
\ifx\@kernlena\@minussign
  \oldkerna
\else
  \kern#2\psunit
\fi
\char\the\c@@slotb%
\ifx\@kernlenb\@minussign
  \oldkernb
\else
  \kern#4\psunit
\fi
\char\the\c@@slotc%
&
```

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.

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

Switch to the legend font.

```
\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 "-".

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

```
        \ifdim\kernlen pt<0pt-\fi
        \ifdim\kernlen pt>0pt+\fi
        \kernlen}%
    }{%
      \ifnum\kernlen=0\relax
      \else
        \textcolor{oldcolor}{\small(%
          \ifdim\kernlen pt<0pt-\fi
          \ifdim\kernlen pt>0pt+\fi
          \kernlen)}%
      \fi
      ~\textcolor{newcolor}{\large
        \ifnum
        \csname kt@kerning@\getpsname{\the\c@@slota}@%
        \getpsname{\the\c@@slotb}\endcsname<0-\fi
        \csname kt@kerning@\getpsname{\the\c@@slota}@%
        \getpsname{\the\c@@slotb}\endcsname
        \makebox[0pt][l]{$^\ast$}%
      }%
    }%
```

Write old kerning to `mtx` file.

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

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

```
  \else
```

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

```
    \ifdim\kernlen pt=0pt
    \else
      \textcolor{oldcolor}{\small(%
        \ifdim\kernlen pt<0pt-\fi
        \ifdim\kernlen pt>0pt+\fi
        \kernlen)}
    \fi
    \textcolor{newcolor}{\large
      \ifnum#2<0-\fi#2%
      \ifthenelse{\equal{\oldkerna}{}}{%
      }{\makebox[0mm][l]{$^\dagger$}}%
    }%
```

Write the new kerning information into the `mtx` file.

```
    \ifthenelse{\equal{\oldkerna}{}}{%
      \writemtxkern{\the\c@@slota}{#2}{\the\c@@slotb}%
    }{}%
  \fi
  &
```

Do the same for the second kerning pair.

```
  \getkern{\char\the\c@@slotb}{\char\the\c@@slotc}%
  \krntst@helpfont
  \ifx\@kernlenb\@minussign
    \ifthenelse{\equal{\oldkernb}{}}{%
```

```
      \textcolor{oldcolor}{\small
        \ifdim\kernlen pt<0pt-\fi
        \ifdim\kernlen pt>0pt+\fi
        \kernlen}%
    }{%
      \ifnum\kernlen=0\relax
      \else
        \textcolor{oldcolor}{\small(%
          \ifdim\kernlen pt<0pt-\fi
          \ifdim\kernlen pt>0pt+\fi
          \kernlen)}%
      \fi
      ~\textcolor{newcolor}{\large
        \ifnum
        \csname kt@kerning@\getpsname{\the\c@@slotb}@%
        \getpsname{\the\c@@slotc}\endcsname<0-\fi
        \csname kt@kerning@\getpsname{\the\c@@slotb}@%
        \getpsname{\the\c@@slotc}\endcsname
        \makebox[0pt][l]{$^\ast$}%
      }%
    }%
    \ifthenelse{\boolean{krntst@writeall}\and\not\equal{\kernlen}{0}}{%
      \writemtxkern[original kerning]{\the\c@@slotb}{%
        \ifdim\kernlen pt>0pt+\fi\kernlen}{\the\c@@slotc}%
    }{}%
  \else
    \ifdim\kernlen pt=0pt
    \else
      \textcolor{oldcolor}{\small(%
        \ifdim\kernlen pt<0pt-\fi
        \ifdim\kernlen pt>0pt+\fi
        \kernlen)}
    \fi
    \textcolor{newcolor}{\large
      \ifnum#4<0-\fi#4%
      \ifthenelse{\equal{\oldkernb}{}}{%
        }{\makebox[0mm][l]{$^\dagger$}}%
    }%
    \ifthenelse{\equal{\oldkernb}{}}{%
      \writemtxkern{\the\c@@slotb}{#4}{\the\c@@slotc}%
    }{}%
  \fi
  &
```

Print the example with natural kerning.

```
  \char\the\c@@slota\char\the\c@@slotb
  \krntst@example
  \char\the\c@@slotb\char\the\c@@slotc
  &
```

Print the example with new kerning.

```
  \char\the\c@@slota%
  \ifx\@kernlena\@minussign
  \else
    \kern#2\psunit
```

```
  \fi
  \char\the\c@@slotb\krntst@example\char\the\c@@slotb
  \ifx\@kernlenb\@minussign
  \else
    \kern#4\psunit
  \fi
  \char\the\c@@slotc%
  &
```

Switch to legend font for the comments that may appear.

```
  \krntst@helpfont\ignorespaces
}
```

`kerntable` The kerning table environment.

```
\newenvironment{kerntable}{%
```

Switch to the tested font.

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

Start a `longtable` environment for the kerning samples.

```
  \begin{longtable}[l]{@{}lll@{~}l@{~}l@{~}rrlll@{}}
```

Type the header of the table.

```
    \krntst@helpfont slot&
    \krntst@helpfont name&
    \krntst@helpfont orig&
    \krntst@helpfont new&
    \krntst@helpfont both&
    \krntst@helpfont k.\,1&
    \krntst@helpfont k.\,2&
    \krntst@helpfont orig.&
    \krntst@helpfont new&
    \krntst@helpfont comment\\
    \endhead
```

Repeat it as footer.

```
    \krntst@helpfont slot&
    \krntst@helpfont name&
    \krntst@helpfont orig&
    \krntst@helpfont new&
    \krntst@helpfont both&
    \krntst@helpfont k.\,1&
    \krntst@helpfont k.\,2&
    \krntst@helpfont orig.&
    \krntst@helpfont new&
    \krntst@helpfont comment\\
    \endfoot
  }{%
```

And now the end of the table.

```
  \end{longtable}%
  \ignorespacesafterend
}
```

\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).

```
\newif\if@tempswb
\newcommand\writemtxkern[4][\@empty]{%
```

Store the glyph names of both glyphs in \@firstglyphname and \@secondglyphname.

```
  \expandafter\@saveslotnumber#2\@empty
  \edef\@firstglyphname{\getpsname{\c@@glyphslot}}%
  \expandafter\@saveslotnumber#4\@empty
  \edef\@secondglyphname{\getpsname{\c@@glyphslot}}%
```

Test if a comment has been given.

```
  \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.

```
    \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.

```
    \ifthenelse{\equal{\rightkern}{\@empty}}{%
      \edef\textright{./.}%
      \def\rightkern{@firstglyphname}%
    }{%
      \edef\textright{\expandafter\@getclassname\rightkern\@empty}%
    }%
```

Get the corresponding glyph class for the second character and save it in \leftkern. If none, \leftkern is set to \@empty.

```
    \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 \leftright for the mtx file.

```
    \ifthenelse{\equal{\leftkern}{\@empty}}{%
      \edef\textleft{./.}%
      \def\leftkern{@secondglyphname}%
    }{%
      \edef\textleft{\expandafter\@getclassname\leftkern\@empty}%
    }%
```

Set the kernig data for all kerning pairs that can be found in both glyph classes \rightkern and \leftkern.

```
    \@tempswbtrue
    \@forallinclass{\rightkern}{first}{%
      \@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.

```
        \if@tempswb
          \protected@write\mtxfile{}{%
            \string\setkern
            {\first}{\second}{#3}%
            \@percentchar\space\space\textright\space-- \textleft
          }% protected@write
          \@tempswbfalse
        \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.

```
        \setcounter{@tmpscale}{#3*\first@scaling*\second@scaling/1000000}%
        \typeout{\first-\second: \the@tmpscale}%
        \expandafter\xdef\csname kt@kerning@\first @\second\endcsname{%
          \the@tmpscale}%
%       \expandafter\xdef\csname kt@kerning@\first @\second\endcsname{#3}%
      }% forallinclass second
    }% forallinclass first
  }{%
```

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

```
    \protected@write\mtxfile{}{%
      \string\setkern
      {\@firstglyphname}{\@secondglyphname}{#3}%
      \@percentchar\space\space #1%
    }%
```

Nevertheless, generate the command for testing on conflicting values.

```
    \expandafter\xdef
      \csname kt@kerning@\@firstglyphname@\@secondglyphname\endcsname{#3}%
  }%
}%
```

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.

```
\newcounter{@tmpscale}
\newcounter{@firstscale}
\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.

```
  \AtBeginDocument{%
```

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

```
    \@ifundefined{glyphclasslist@#1}{%
```

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

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

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

```
      \begingroup
      \@tempswatrue
      \forallclasses{#1}{@tmpcls}{%
        \ifthenelse{\equal{\@tmpcls}{glyphclass@#1@#2}}{%
          \@tempswafalse
```

```
      }{}%
    }%
```

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

```
    \if@tempswa
      \expandafter\xdef\csname glyphclasslist@#1\endcsname{%
        \csname glyphclasslist@#1\endcsname,glyphclass@#1@#2}%
    \fi
    \endgroup
  }%
```

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".

```
  \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.

```
  \edef\@tempa{#3}%
  \@tempswbtrue
  \@forallinclass{@tempa}{@tmpglyph}{%
    \saveslotnumber{@glyphslot}{\@tmpglyph}%
    \ifnum\the\c@@glyphslot<0%
      \ClassError{kerntest}{Used unknown glyph `\@tmpglyph'}{%
        You may have misspelled the glyph or you have taken an invalid
        number.
        This incorrect glyph is ignored.}%
      \edef\thisglyphname{???}%
    \else
      \edef\thisglyphname{\getpsname{\the\c@@glyphslot}}%
      \if@tempswb
        \global\@tempswbfalse
        \setcounter{@firstscale}{\@tmpglyph@scaling}%
      \fi
      \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.

```
      \@tempswatrue
      \forallclasses{#1}{@tmpcls}{%
        \@ifglyphinclass{\@tmpcls}{\@tmpglyph}{%
          \@tempswafalse
          \ClassError{kerntest}{Glyph `\@tmpglyph'
            (`\thisglyphname',\MessageBreak
            glyph class #1/#2) already\MessageBreak
            in glyph
            class (\expandafter\@getclassname\@tmpcls\@empty)%
          }{%
            Each glyph may only be once in one glyph class for
            each side.
          }%
        }{}%
      }%
```

Append this glyph to the current glyph list.

```
      \if@tempswa
        \ifthenelse{\equal{\csname glyphclass@#1@#2\endcsname}{}}{%
          \expandafter\edef\csname glyphclass@#1@#2\endcsname{%
            \thisglyphname[\the@tmpscale]}%
          \edef\firstglyphinclass{\thisglyphname}%
        }{%
          \expandafter\edef\csname glyphclass@#1@#2\endcsname{%
            \csname glyphclass@#1@#2\endcsname,%
            \thisglyphname[\the@tmpscale]}%
        }%
```

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

```
        \expandafter\edef
        \csname glyphclass@glyph@#1@\thisglyphname\endcsname{%
          glyphclass@#1@#2}%
      \fi
    \fi
  }%
```

Some feedback.

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

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

```
\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.

```
  \AtBeginDocument{%
    \@tempswatrue
    \forallclasses{#1}{@tmpcls}{%
      \ifthenelse{\equal{\@tmpcls}{glyphclass@#1@#2}}{%
        \ClassError{kerntest}{Class `#1/#2' already exists}{%
          The command is ignored.}%
        \@tempswafalse
      }{}%
    }%
    \if@tempswa
      \defglyphclass{#1}{#2}{#3}%
    \fi
  }%
}
```

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

```
\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.

```
  \AtBeginDocument{%
```

```
    \@tempswafalse
    \forallclasses{#1}{@tmpcls}{%
      \ifthenelse{\equal{\@tmpcls}{glyphclass@#1@#2}}{%
        \@tempswatrue
      }{}%
    }%
    \if@tempswa
      \defglyphclass{#1}{#2}{#3}%
    \else
      \ClassError{kerntest}{Class `#1/#2' does not exists}{%
        The command is ignored.}%
    \fi
  }%
}
```

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

```
\newcommand\provideglyphclass[3]{%
```

Test if this glyph class already exists. If not, call `\defglyphclass` to save the new glyph class.

```
  \AtBeginDocument{%
    \@tempswatrue
    \forallclasses{#1}{@tmpcls}{%
      \ifthenelse{\equal{\@tmpcls}{glyphclass@#1@#2}}{%
        \@tempswafalse
      }{}%
    }%
    \if@tempswa
      \defglyphclass{#1}{#2}{#3}%
    \fi
  }%
}
```

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.

```
\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.

```
\newcommand*\getclassofglyph[2]{%
  \expandafter\ifx\csname glyphclass@glyph@#1@#2\endcsname\relax
    \@empty
  \else
    \csname glyphclass@glyph@#1@#2\endcsname
  \fi
}
```

An internal boolean for searching glyph classes.

```
\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.

```
\newcommand\ifglyphinclass[5]{%
  \@ifundefined{glyphclass@#1@#2}{%
    \ClassError{kerntest}{Glyph class #1/#2 not available}{}%
  }{%
    \edef\@tempa{#3}%
    \saveslotnumber{@tempcnta}{\@tempa}%
    \ifnum\the\c@@tempcnta<0%
      \ClassError{kerntest}{Used unknown glyph `#2'}{%
        You may have misspelled the glyph or you have taken an invalid
        number.}%
    \else
      \ifthenelse{\equal{glyphclass@#1@#2}{%
          \getclassofglyph{#1}{\getpsname{\the\c@@tempcnta}}}}{%
        #4%
      }{%
        #5%
      }%
    \fi
%    \@ifglyphinclass{glyphclass@#1@#2}{#3}{#4}{#5}%
  }%
}
```

\@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.

```
\newcounter{@tempcnta}
\newcommand\@ifglyphinclass[4]{%
  \@ifundefined{#1}{%
    \ClassError{kerntest}{Glyph class #1 not available}{}%
  }{%
```

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

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

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

```
    \edef\@tempa{#2}%
```

Find out if the glyph is valid.

```
    \saveslotnumber{@tempcnta}{\@tempa}%
    \ifnum\the\c@@tempcnta<0\relax
      \ClassError{kerntest}{Used unknown glyph `#2'}{%
        You may have misspelled the glyph or you have taken an invalid
        number.}%
    \else
```

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

```
      \ifthenelse{\equal{#1}{%
          \getclassofglyph{\krntst@side}{\getpsname{\the\c@@tempcnta}}}}{%
        #3%
      }{%
```

```
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     }%
```

```
  }%
}
```

`\@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 (`\`⟨*#3*⟩). Store the rest of the glyph class in `\`⟨*#3*⟩`@rest` for later work on it.

```
\def\@@parseglyphname#1[#2]#3\@empty#4{%
  \expandafter\edef\csname#4\endcsname{#1}%
  \expandafter\xdef\csname#4@scaling\endcsname{#2}%
%  \typeout{--> '#1' mit [\csname #4@scaling\endcsname], ignoriert: '#3', Name: '#4'}%
}
\def\@nextglyphinclass#1,#2\@empty#3{%
  \expandafter\edef\csname#3@rest\endcsname{#2}%
  \expandafter\@@parseglyphname#1[1000]\@empty{#3}%
}
```

`\forallclasses` Syntax: `\forallclasses{`⟨*side*⟩`}{`⟨*glyph class*⟩`}{`⟨*action*⟩`}`

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

```
\newcommand\forallclasses[3]{%
  \renewcommand\stoploop[1][#2]{%
    \expandafter\edef\csname ##1\endcsname{}%
  }%
  \begingroup\expandafter\expandafter\expandafter\endgroup
  \expandafter\ifx\csname glyphclasslist@#1\endcsname\relax
  \edef\kt@tempa{}%
  \else
    \edef\kt@tempa{\csname glyphclasslist@#1\endcsname}%
  \fi
  \expandafter\@nextclass\kt@tempa,\@empty{#2}%
  \whiledo{\not\equal{\csname #2\endcsname}{}}{%
    #3%
    \ifthenelse{\equal{\csname #2\endcsname}{}}{%
    }{%
      \edef\kt@tempa{\csname #2@rest\endcsname}%
      \expandafter\@nextclass\kt@tempa,\@empty{#2}%
    }%
  }%
}
```

`\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.

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

`\@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 (`\`⟨*#3*⟩). Store the rest of the glyph-class list in `\`⟨*#3*⟩`@rest` for later work on it.

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

```
  \expandafter\edef\csname #3@rest\endcsname{#2}%
  \expandafter\edef\csname #3\endcsname{#1}%
}
```

6.1.2 Extra commands for special encodings

Setup command for the different encodings.

```
\newcommand\encodingsetup[1]{%
  \AtBeginDocument{%
    \typeout{Setup for font encoding.
      This differs from encoding to encoding.}%
    \setkeys{krnenc}{#1}%
  }%
}
\@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.

```
\AtEndDocument{%
```

First, write the kerning data for the glyph classes.

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

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

```
  \def\@writeclass#1{%
    \forallclasses{#1}{@tmpcls}{%
      \typeout{\space\space for class
        \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.

```
      \@tempswatrue
      \@forallinclass{\@tmpcls}{@tmpglyph}{%
        \if@tempswa
          \edef\firstglyphinclass{\@tmpglyph}%
          \@tempswafalse
        \else
```

For the other members of this class, write the `\set`⟨*side*⟩`kerning` commands to the `mtx` file.

```
          \protected@write\mtxfile{}{%
            \string\set#1kerning%
            {\@tmpglyph}{\firstglyphinclass}{\@tmpglyph@scaling}%
            \@percentchar\space\space
            \expandafter\@getclassname\@tmpcls\@empty
          }%
        \fi
      }%
    }%
```

```
  }%
```
The sides `left` and `right` are defined.
```
  \@writeclass{left}%
  \protected@write\mtxfile{}{\@percentchar\@percentchar}%
  \@writeclass{right}%
```
Write a real footer.
```
  \protected@write\mtxfile{}{\@percentchar\@percentchar}%
  \protected@write\mtxfile{}{\string\endmetrics}%
  \protected@write\mtxfile{}{\@percentchar\@percentchar}%
  \protected@write\mtxfile{}{\@percentchar\@percentchar\space
    End of file `\mtxfilename.mtx'.}%
  \closeout\mtxfile
  \typeout{^^JWritten mtx file `\mtxfilename.mtx'^^J}%
}
```
⟨/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`.

⟨∗mtx & t1⟩
```
\makeatletter
```
Set options to switch to other font shapes.
```
\define@key{krnenc}{ligaturing}[1]{%
  \typeout{T1 encoding: ligaturing = #1}%
  \ifnum#1<0\relax
    \expandafter\def\csname krntst@T1@027\endcsname{.notdef.027}%
    \expandafter\def\csname krntst@T1@028\endcsname{.notdef.028}%
    \expandafter\def\csname krntst@T1@029\endcsname{.notdef.029}%
    \expandafter\def\csname krntst@T1@030\endcsname{.notdef.030}%
    \expandafter\def\csname krntst@T1@031\endcsname{.notdef.031}%
    \expandafter\def\csname krntst@T1@156\endcsname{.notdef.156}%
    \expandafter\def\csname krntst@T1@188\endcsname{.notdef.188}%
  \else
    \expandafter\def\csname krntst@T1@027\endcsname{ff}%
    \expandafter\def\csname krntst@T1@028\endcsname{fi}%
    \expandafter\def\csname krntst@T1@029\endcsname{fl}%
    \expandafter\def\csname krntst@T1@030\endcsname{ffi}%
    \expandafter\def\csname krntst@T1@031\endcsname{ffl}%
    \expandafter\def\csname krntst@T1@156\endcsname{IJ}%
    \expandafter\def\csname krntst@T1@188\endcsname{ij}%
  \fi
  \ifnum#1<-1\relax
    \expandafter\def\csname krntst@T1@073\endcsname{.notdef.073}%
    \expandafter\def\csname krntst@T1@099\endcsname{.notdef.099}%
    \expandafter\def\csname krntst@T1@102\endcsname{.notdef.102}%
    \expandafter\def\csname krntst@T1@105\endcsname{.notdef.105}%
```

```
    \expandafter\def\csname krntst@T1@115\endcsname{.notdef.115}%
  \else
    \expandafter\def\csname krntst@T1@073\endcsname{I}%
    \expandafter\def\csname krntst@T1@099\endcsname{c}%
    \expandafter\def\csname krntst@T1@102\endcsname{f}%
    \expandafter\def\csname krntst@T1@105\endcsname{i}%
    \expandafter\def\csname krntst@T1@115\endcsname{s}%
  \fi
}
\typeout{^^JValid values for T1 encoding:}%
\typeout{ligaturing: -2, -1, 0, 1}%
\typeout{Defaults for T1 encoding:}%
\setkeys{krnenc}{ligaturing=1}%
\typeout{}%
\makeatother
```

Now, set `\getpsname`.

```
\newcommand\getpsname[1]{%
  \ifcase#1%
  grave\or% 000
  acute\or% 001
  circumflex\or% 002
  tilde\or% 003
  dieresis\or% 004
  hungarumlaut\or% 005
  ring\or% 006
  caron\or% 007
  breve\or% 008
  macron\or% 009
  dotaccent\or% 010
  cedilla\or% 011
  ogonek\or% 012
  quotesinglbase\or% 013
  guilsinglleft\or% 014
  guilsinglright\or% 015
  quotedblleft\or% 016
  quotedblright\or% 017
  quotedblbase\or% 018
  guillemotleft\or% 019
  guillemotright\or% 020
  rangedash\or% 021
  punctdash\or% 022
  compwordmark\or% 023
  perthousandzero\or% 024
  dotlessi\or% 025
  dotlessj\or% 026
  \csname krntst@T1@027\endcsname\or % 027
  \csname krntst@T1@028\endcsname\or % 028
  \csname krntst@T1@029\endcsname\or % 029
  \csname krntst@T1@030\endcsname\or % 030
  \csname krntst@T1@031\endcsname\or % 031
  visiblespace\or% 032
  exclam\or% 033
  quotedbl\or% 034
  numbersign\or% 035
```

```
dollar\or% 036
percent\or% 037
ampersand\or% 038
quoteright\or% 039
parenleft\or% 040
parenright\or% 041
asterisk\or% 042
plus\or% 043
comma\or% 044
hyphen\or% 045
period\or% 046
slash\or% 047
zero\or% 048
one\or% 049
two\or% 050
three\or% 051
four\or% 052
five\or% 053
six\or% 054
seven\or% 055
eight\or% 056
nine\or% 057
colon\or% 058
semicolon\or% 059
less\or% 060
equal\or% 061
greater\or% 062
question\or% 063
at\or% 064
A\or% 065
B\or% 066
C\or% 067
D\or% 068
E\or% 069
F\or% 070
G\or% 071
H\or% 072
\csname krntst@T1@073\endcsname\or % 073
J\or% 074
K\or% 075
L\or% 076
M\or% 077
N\or% 078
O\or% 079
P\or% 080
Q\or% 081
R\or% 082
S\or% 083
T\or% 084
U\or% 085
V\or% 086
W\or% 087
X\or% 088
Y\or% 089
```

```
Z\or% 090
bracketleft\or% 091
backslash\or% 092
bracketright\or% 093
asciicircum\or% 094
underscore\or% 095
quoteleft\or% 096
a\or% 097
b\or% 098
\csname krntst@T1@099\endcsname\or % 099
d\or% 100
e\or% 101
\csname krntst@T1@102\endcsname\or % 102
g\or% 103
h\or% 104
\csname krntst@T1@105\endcsname\or % 105
j\or% 106
k\or% 107
l\or% 108
m\or% 109
n\or% 110
o\or% 111
p\or% 112
q\or% 113
r\or% 114
\csname krntst@T1@115\endcsname\or % 115
t\or% 116
u\or% 117
v\or% 118
w\or% 119
x\or% 120
y\or% 121
z\or% 122
braceleft\or% 123
bar\or% 124
braceright\or% 125
asciitilde\or% 126
hyphenchar\or% 127
Abreve\or% 128
Aogonek\or% 129
Cacute\or% 130
Ccaron\or% 131
Dcaron\or% 132
Ecaron\or% 133
Eogonek\or% 134
Gbreve\or% 135
Lacute\or% 136
Lcaron\or% 137
Lslash\or% 138
Nacute\or% 139
Ncaron\or% 140
Ng\or% 141
Ohungarumlaut\or% 142
Racute\or% 143
```

```
Rcaron\or% 144
Sacute\or% 145
Scaron\or% 146
Scedilla\or% 147
Tcaron\or% 148
Tcedilla\or% 149
Uhungarumlaut\or% 150
Uring\or% 151
Ydieresis\or% 152
Zacute\or% 153
Zcaron\or% 154
Zdotaccent\or% 155
\csname krntst@T1@156\endcsname\or % 156
Idotaccent\or% 157
dbar\or% 158
section\or% 159
abreve\or% 160
aogonek\or% 161
cacute\or% 162
ccaron\or% 163
dcaron\or% 164
ecaron\or% 165
eogonek\or% 166
gbreve\or% 167
lacute\or% 168
lcaron\or% 169
lslash\or% 170
nacute\or% 171
ncaron\or% 172
ng\or% 173
ohungarumlaut\or% 174
racute\or% 175
rcaron\or% 176
sacute\or% 177
scaron\or% 178
scedilla\or% 179
tcaron\or% 180
tcedilla\or% 181
uhungarumlaut\or% 182
uring\or% 183
ydieresis\or% 184
zacute\or% 185
zcaron\or% 186
zdotaccent\or% 187
\csname krntst@T1@188\endcsname\or % 188
exclamdown\or% 189
questiondown\or% 190
sterling\or% 191
Agrave\or% 192
Aacute\or% 193
Acircumflex\or% 194
Atilde\or% 195
Adieresis\or% 196
Aring\or% 197
```

```
AE\or% 198
Ccedilla\or% 199
Egrave\or% 200
Eacute\or% 201
Ecircumflex\or% 202
Edieresis\or% 203
Igrave\or% 204
Iacute\or% 205
Icircumflex\or% 206
Idieresis\or% 207
Eth\or% 208
Ntilde\or% 209
Ograve\or% 210
Oacute\or% 211
Ocircumflex\or% 212
Otilde\or% 213
Odieresis\or% 214
OE\or% 215
Oslash\or% 216
Ugrave\or% 217
Uacute\or% 218
Ucircumflex\or% 219
Udieresis\or% 220
Yacute\or% 221
Thorn\or% 222
SS\or% 223
agrave\or% 224
aacute\or% 225
acircumflex\or% 226
atilde\or% 227
adieresis\or% 228
aring\or% 229
ae\or% 230
ccedilla\or% 231
egrave\or% 232
eacute\or% 233
ecircumflex\or% 234
edieresis\or% 235
igrave\or% 236
iacute\or% 237
icircumflex\or% 238
idieresis\or% 239
eth\or% 240
ntilde\or% 241
ograve\or% 242
oacute\or% 243
ocircumflex\or% 244
otilde\or% 245
odieresis\or% 246
oe\or% 247
oslash\or% 248
ugrave\or% 249
uacute\or% 250
ucircumflex\or% 251
```

```
  udieresis\or% 252
  yacute\or% 253
  thorn\or% 254
  germandbls\fi% 255
}
⟨/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.

```
⟨*mtx & ts1⟩
\newcommand\getpsname[1]{%
  \ifcase#1%
  capitalgrave\or% 000
  capitalacute\or% 001
  capitalcircumflex\or% 002
  capitaltilde\or% 003
  capitaldieresis\or% 004
  capitalhungarumlaut\or% 005
  capitalring\or% 006
  capitalcaron\or% 007
  capitalbreve\or% 008
  capitalmacron\or% 009
  capitaldotaccent\or% 010
  cedilla\or% 011
  ogonek\or% 012
  quotesinglbase\or% 013
  .notdef.014\or% 014
  .notdef.015\or% 015
  .notdef.016\or% 016
  .notdef.017\or% 017
  quotedblbase\or% 018
  .notdef.019\or% 019
  .notdef.020\or% 020
  twelveudash\or% 021
  threequartersemdash\or% 022
  capitalcompwordmark\or% 023
  arrowleft\or% 024
  arrowright\or% 025
  tieaccentlowercase\or% 026
  tieaccentcapital\or% 027
  newtieaccentlowercase\or% 028
  newtieaccentcapital\or% 029
  ascendercompwordmark\or% 030
  blank\or% 031
  .notdef.032\or% 032
  .notdef.033\or% 033
  .notdef.034\or% 034
  .notdef.035\or% 035
  dollar\or% 036
  .notdef.037\or% 037
  .notdef.038\or% 038
```

```
quotesingle\or% 039
.notdef.040\or% 040
.notdef.041\or% 041
asteriskcentered\or% 042
.notdef.043\or% 043
comma\or% 044
hyphendbl\or% 045
period\or% 046
fraction\or% 047
zerooldstyle\or% 048
oneoldstyle\or% 049
twooldstyle\or% 050
threeoldstyle\or% 051
fouroldstyle\or% 052
fiveoldstyle\or% 053
sixoldstyle\or% 054
sevenoldstyle\or% 055
eightoldstyle\or% 056
nineoldstyle\or% 057
.notdef.058\or% 058
.notdef.059\or% 059
angbracketleft\or% 060
minus\or% 061
angbracketright\or% 062
.notdef.063\or% 063
.notdef.064\or% 064
.notdef.065\or% 065
.notdef.066\or% 066
.notdef.067\or% 067
.notdef.068\or% 068
.notdef.069\or% 069
.notdef.070\or% 070
.notdef.071\or% 071
.notdef.072\or% 072
.notdef.073\or% 073
.notdef.074\or% 074
.notdef.075\or% 075
.notdef.076\or% 076
Omegainv\or% 077
.notdef.078\or% 078
bigcircle\or% 079
.notdef.080\or% 080
.notdef.081\or% 081
.notdef.082\or% 082
.notdef.083\or% 083
.notdef.084\or% 084
.notdef.085\or% 085
.notdef.086\or% 086
Omega\or% 087
.notdef.088\or% 088
.notdef.089\or% 089
.notdef.090\or% 090
openbracketleft\or% 091
.notdef.092\or% 092
```

```
openbracketright\or% 093
arrowup\or% 094
arrowdown\or% 095
asciigrave\or% 096
.notdef.097\or% 097
born\or% 098
divorced\or% 099
died\or% 100
.notdef.101\or% 101
.notdef.102\or% 102
.notdef.103\or% 103
.notdef.104\or% 104
.notdef.105\or% 105
.notdef.106\or% 106
.notdef.107\or% 107
leaf\or% 108
married\or% 109
musicalnote\or% 110
.notdef.111\or% 111
.notdef.112\or% 112
.notdef.113\or% 113
.notdef.114\or% 114
.notdef.115\or% 115
.notdef.116\or% 116
.notdef.117\or% 117
.notdef.118\or% 118
.notdef.119\or% 119
.notdef.120\or% 120
.notdef.121\or% 121
.notdef.122\or% 122
.notdef.123\or% 123
.notdef.124\or% 124
.notdef.125\or% 125
tildelow\or% 126
hyphendblchar\or% 127
asciibreve\or% 128
asciicaron\or% 129
asciiacutedbl\or% 130
asciigravedbl\or% 131
dagger\or% 132
daggerdbl\or% 133
bardbl\or% 134
perthousand\or% 135
bullet\or% 136
centigrade\or% 137
dollaroldstyle\or% 138
centoldstyle\or% 139
florin\or% 140
colonmonetary\or% 141
won\or% 142
naira\or% 143
guarani\or% 144
peso\or% 145
lira\or% 146
```

```
recipe\or% 147
interrobang\or% 148
interrobangdown\or% 149
dong\or% 150
trademark\or% 151
pertenthousand\or% 152
pilcrow\or% 153
baht\or% 154
numero\or% 155
discount\or% 156
estimated\or% 157
openbullet\or% 158
servicemark\or% 159
quillbracketleft\or% 160
quillbracketright\or% 161
cent\or% 162
sterling\or% 163
currency\or% 164
yen\or% 165
brokenbar\or% 166
section\or% 167
asciidieresis\or% 168
copyright\or% 169
ordfeminine\or% 170
copyleft\or% 171
logicalnot\or% 172
circledP\or% 173
registered\or% 174
asciimacron\or% 175
degree\or% 176
plusminus\or% 177
twosuperior\or% 178
threesuperior\or% 179
asciiacute\or% 180
mu\or% 181
paragraph\or% 182
periodcentered\or% 183
referencemark\or% 184
onesuperior\or% 185
ordmasculine\or% 186
radical\or% 187
onequarter\or% 188
onehalf\or% 189
threequarters\or% 190
euro\or% 191
.notdef.192\or% 192
.notdef.193\or% 193
.notdef.194\or% 194
.notdef.195\or% 195
.notdef.196\or% 196
.notdef.197\or% 197
.notdef.198\or% 198
.notdef.199\or% 199
.notdef.200\or% 200
```

```
  .notdef.201\or% 201
  .notdef.202\or% 202
  .notdef.203\or% 203
  .notdef.204\or% 204
  .notdef.205\or% 205
  .notdef.206\or% 206
  .notdef.207\or% 207
  .notdef.208\or% 208
  .notdef.209\or% 209
  .notdef.210\or% 210
  .notdef.211\or% 211
  .notdef.212\or% 212
  .notdef.213\or% 213
  multiply\or% 214
  .notdef.215\or% 215
  .notdef.216\or% 216
  .notdef.217\or% 217
  .notdef.218\or% 218
  .notdef.219\or% 219
  .notdef.220\or% 220
  .notdef.221\or% 221
  .notdef.222\or% 222
  .notdef.223\or% 223
  .notdef.224\or% 224
  .notdef.225\or% 225
  .notdef.226\or% 226
  .notdef.227\or% 227
  .notdef.228\or% 228
  .notdef.229\or% 229
  .notdef.230\or% 230
  .notdef.231\or% 231
  .notdef.232\or% 232
  .notdef.233\or% 233
  .notdef.234\or% 234
  .notdef.235\or% 235
  .notdef.236\or% 236
  .notdef.237\or% 237
  .notdef.238\or% 238
  .notdef.239\or% 239
  .notdef.240\or% 240
  .notdef.241\or% 241
  .notdef.242\or% 242
  .notdef.243\or% 243
  .notdef.244\or% 244
  .notdef.245\or% 245
  divide\or% 246
  .notdef.247\or% 247
  .notdef.248\or% 248
  .notdef.249\or% 249
  .notdef.250\or% 250
  .notdef.251\or% 251
  .notdef.252\or% 252
  .notdef.253\or% 253
  .notdef.254\or% 254
```

```
  .notdef.255\fi% 255
}
⟨/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.

```
⟨*mtx & ot1⟩
\makeatletter
```

Set options to switch to other font shapes.

```
\define@key{krnenc}{ligaturing}[2]{%
  \typeout{OT1 encoding: ligaturing = #1}%
  \ifnum#1=2\relax
    \expandafter\def\csname krntst@OT1@011\endcsname{ff}%
    \expandafter\def\csname krntst@OT1@012\endcsname{fi}%
    \expandafter\def\csname krntst@OT1@013\endcsname{fl}%
    \expandafter\def\csname krntst@OT1@014\endcsname{ffi}%
    \expandafter\def\csname krntst@OT1@015\endcsname{ffl}%
    \expandafter\def\csname krntst@OT1@060\endcsname{exclamdown}%
    \expandafter\def\csname krntst@OT1@062\endcsname{questiondown}%
  \else
    \expandafter\def\csname krntst@OT1@011\endcsname{arrowup}%
    \expandafter\def\csname krntst@OT1@012\endcsname{arrowdown}%
    \expandafter\def\csname krntst@OT1@013\endcsname{quotesingle}%
    \expandafter\def\csname krntst@OT1@014\endcsname{exclamdown}%
    \expandafter\def\csname krntst@OT1@015\endcsname{questiondown}%
    \expandafter\def\csname krntst@OT1@060\endcsname{less}%
    \expandafter\def\csname krntst@OT1@062\endcsname{greater}%
  \fi
  \ifnum#1=0\relax
    \expandafter\def\csname krntst@OT1@032\endcsname{visiblespace}%
    \expandafter\def\csname krntst@OT1@034\endcsname{quotedbl}%
    \expandafter\def\csname krntst@OT1@092\endcsname{backslash}%
    \expandafter\def\csname krntst@OT1@095\endcsname{underscore}%
    \expandafter\def\csname krntst@OT1@123\endcsname{braceleft}%
    \expandafter\def\csname krntst@OT1@124\endcsname{bar}%
    \expandafter\def\csname krntst@OT1@125\endcsname{braceright}%
  \else
    \expandafter\def\csname krntst@OT1@032\endcsname{lslashslash}%
    \expandafter\def\csname krntst@OT1@034\endcsname{quotedblright}%
    \expandafter\def\csname krntst@OT1@092\endcsname{quotedblleft}%
    \expandafter\def\csname krntst@OT1@095\endcsname{dotaccent}%
    \expandafter\def\csname krntst@OT1@123\endcsname{rangedash}%
    \expandafter\def\csname krntst@OT1@124\endcsname{punctdash}%
    \expandafter\def\csname krntst@OT1@125\endcsname{hungarumlaut}%
  \fi
}
```

```
\define@key{krnenc}{italicizing}[true]{%
  \typeout{OT1 encoding: italicizing = #1}%
  \csname if#1\endcsname
    \expandafter\def\csname krntst@OT1@036\endcsname{sterling}%
  \else
    \expandafter\def\csname krntst@OT1@036\endcsname{dollar}
  \fi
}
\typeout{^^JValid values for OT1 encoding:}%
\typeout{ligaturing: 0, 1, 2}%
\typeout{italicizing: false, true}%
\typeout{Defaults for OT1 encoding:}%
\setkeys{krnenc}{ligaturing=2,italicizing=false}%
\typeout{}%
\makeatother
```

Now, set `\getpsname`.

```
\newcommand\getpsname[1]{%
  \ifcase#1%
  Gamma\or % 000
  Delta\or % 001
  Theta\or % 002
  Lambda\or % 003
  Xi\or % 004
  Pi\or % 005
  Sigma\or % 006
  Upsilon\or % 007
  Phi\or % 008
  Psi\or % 009
  Omega\or % 010
  \csname krntst@OT1@011\endcsname\or % 011
  \csname krntst@OT1@012\endcsname\or % 012
  \csname krntst@OT1@013\endcsname\or % 013
  \csname krntst@OT1@014\endcsname\or % 014
  \csname krntst@OT1@015\endcsname\or % 015
  dotlessi\or % 016
  dotlessj\or % 017
  grave\or % 018
  acute\or % 019
  caron\or % 020
  breve\or % 021
  macron\or % 022
  ringfitted\or % 023
  cedilla\or % 024
  germandbls\or % 025
  ae\or % 026
  oe\or % 027
  oslash\or % 028
  AE\or % 029
  OE\or % 030
  Oslash\or % 031
  \csname krntst@OT1@032\endcsname\or % 032
  exclam\or % 033
  \csname krntst@OT1@034\endcsname\or % 034
  numbersign\or % 035
```

```
\csname krntst@OT1@036\endcsname\or % 036
percent\or % 037
ampersand\or % 038
quoteright\or % 039
parenleft\or % 040
parenright\or % 041
asterisk\or % 042
plus\or % 043
comma\or % 044
hyphen\or % 045
period\or % 046
slash\or % 047
zero\or % 048
one\or % 049
two\or % 050
three\or % 051
four\or % 052
five\or % 053
six\or % 054
seven\or % 055
eight\or % 056
nine\or % 057
colon\or % 058
semicolon\or % 059
\csname krntst@OT1@060\endcsname\or % 060
equal\or % 061
\csname krntst@OT1@062\endcsname\or % 062
question\or % 063
at\or % 064
A\or % 065
B\or % 066
C\or % 067
D\or % 068
E\or % 069
F\or % 070
G\or % 071
H\or % 072
I\or % 073
J\or % 074
K\or % 075
L\or % 076
M\or % 077
N\or % 078
O\or % 079
P\or % 080
Q\or % 081
R\or % 082
S\or % 083
T\or % 084
U\or % 085
V\or % 086
W\or % 087
X\or % 088
Y\or % 089
```

```
Z\or % 090
bracketleft\or % 091
\csname krntst@OT1@092\endcsname\or % 092
bracketright\or % 093
circumflex\or % 094
\csname krntst@OT1@095\endcsname\or % 095
quoteleft\or % 096
a\or % 097
b\or % 098
c\or % 099
d\or % 100
e\or % 101
f\or % 102
g\or % 103
h\or % 104
i\or % 105
j\or % 106
k\or % 107
l\or % 108
m\or % 109
n\or % 110
o\or % 111
p\or % 112
q\or % 113
r\or % 114
s\or % 115
t\or % 116
u\or % 117
v\or % 118
w\or % 119
x\or % 120
y\or % 121
z\or % 122
\csname krntst@OT1@123\endcsname\or % 123
\csname krntst@OT1@124\endcsname\or % 124
\csname krntst@OT1@125\endcsname\or % 125
tilde\or % 126
dieresis\or % 127
.notdef.128\or % 128
.notdef.129\or % 129
.notdef.130\or % 130
.notdef.131\or % 131
.notdef.132\or % 132
.notdef.133\or % 133
.notdef.134\or % 134
.notdef.135\or % 135
.notdef.136\or % 136
.notdef.137\or % 137
Lslash\or % 138
.notdef.139\or % 139
.notdef.140\or % 140
.notdef.141\or % 141
.notdef.142\or % 142
.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
```

```
  .notdef.252\or % 252
  .notdef.253\or % 253
  .notdef.254\or % 254
  .notdef.255\fi % 255
}
```
⟨/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.

```
⟨*mtx & t2a⟩
\newcommand\getpsname[1]{%
  \ifcase#1%
  grave\or% 000
  acute\or% 001
  circumflex\or% 002
  tilde\or% 003
  dieresis\or% 004
  hungarumlaut\or% 005
  ring\or% 006
  caron\or% 007
  breve\or% 008
  macron\or% 009
  dotaccent\or% 010
  cedilla\or% 011
  ogonek\or% 012
  CYRpalochka\or% 013
  angleleft\or% 014
  angleright\or% 015
  quotedblleft\or% 016
  quotedblright\or% 017
  cyrflex\or% 018
  dblgrave\or% 019
  cyrbreve\or% 020
  rangedash\or% 021
  cyrdash\or% 022
  compwordmark\or% 023
  perthousandzero\or% 024
  dotlessi\or% 025
  dotlessj\or% 026
  ff\or% 027
  fi\or% 028
  fl\or% 029
  ffi\or% 030
  ffl\or% 031
  visiblespace\or% 032
  exclam\or% 033
  quotedbl\or% 034
  numbersign\or% 035
  dollar\or% 036
  percent\or% 037
  ampersand\or% 038
```

```
quoteright\or% 039
parenleft\or% 040
parenright\or% 041
asterisk\or% 042
plus\or% 043
comma\or% 044
hyphen\or% 045
period\or% 046
slash\or% 047
zero\or% 048
one\or% 049
two\or% 050
three\or% 051
four\or% 052
five\or% 053
six\or% 054
seven\or% 055
eight\or% 056
nine\or% 057
colon\or% 058
semicolon\or% 059
less\or% 060
equal\or% 061
greater\or% 062
question\or% 063
at\or% 064
A\or% 065
B\or% 066
C\or% 067
D\or% 068
E\or% 069
F\or% 070
G\or% 071
H\or% 072
I\or% 073
J\or% 074
K\or% 075
L\or% 076
M\or% 077
N\or% 078
O\or% 079
P\or% 080
Q\or% 081
R\or% 082
S\or% 083
T\or% 084
U\or% 085
V\or% 086
W\or% 087
X\or% 088
Y\or% 089
Z\or% 090
bracketleft\or% 091
backslash\or% 092
```

```
bracketright\or% 093
asciicircum\or% 094
underscore\or% 095
quoteleft\or% 096
a\or% 097
b\or% 098
c\or% 099
d\or% 100
e\or% 101
f\or% 102
g\or% 103
h\or% 104
i\or% 105
j\or% 106
k\or% 107
l\or% 108
m\or% 109
n\or% 110
o\or% 111
p\or% 112
q\or% 113
r\or% 114
s\or% 115
t\or% 116
u\or% 117
v\or% 118
w\or% 119
x\or% 120
y\or% 121
z\or% 122
braceleft\or% 123
bar\or% 124
braceright\or% 125
asciitilde\or% 126
hyphenchar\or% 127
CYRGUP\or% 128
CYRGHCRS\or% 129
CYRDJE\or% 130
CYRTSHE\or% 131
CYRSHHA\or% 132
CYRZHDSC\or% 133
CYRZDSC\or% 134
CYRLJE\or% 135
CYRYI\or% 136
CYRKDSC\or% 137
CYRKBEAK\or% 138
CYRKVCRS\or% 139
CYRAE\or% 140
CYRNDSC\or% 141
CYRNG\or% 142
CYRDZE\or% 143
CYROTLD\or% 144
CYRSDSC\or% 145
CYRUSHRT\or% 146
```

```
CYRY\or% 147
CYRYHCRS\or% 148
CYRHDSC\or% 149
CYRDZHE\or% 150
CYRCHVCRS\or% 151
CYRCHRDSC\or% 152
CYRIE\or% 153
CYRSCHWA\or% 154
CYRNJE\or% 155
CYRYO\or% 156
numero\or% 157
currency\or% 158
section\or% 159
cyrgup\or% 160
cyrghcrs\or% 161
cyrdje\or% 162
cyrtshe\or% 163
cyrshha\or% 164
cyrzhdsc\or% 165
cyrzdsc\or% 166
cyrlje\or% 167
cyryi\or% 168
cyrkdsc\or% 169
cyrkbeak\or% 170
cyrkvcrs\or% 171
cyrae\or% 172
cyrndsc\or% 173
cyrng\or% 174
cyrdze\or% 175
cyrotld\or% 176
cyrsdsc\or% 177
cyrushrt\or% 178
cyry\or% 179
cyryhcrs\or% 180
cyrhdsc\or% 181
cyrdzhe\or% 182
cyrchvcrs\or% 183
cyrchrdsc\or% 184
cyrie\or% 185
cyrschwa\or% 186
cyrnje\or% 187
cyryo\or% 188
quotedblbase\or% 189
guillemotleft\or% 190
guillemotright\or% 191
CYRA\or% 192
CYRB\or% 193
CYRV\or% 194
CYRG\or% 195
CYRD\or% 196
CYRE\or% 197
CYRZH\or% 198
CYRZ\or% 199
CYRI\or% 200
```

```
CYRISHRT\or% 201
CYRK\or% 202
CYRL\or% 203
CYRM\or% 204
CYRN\or% 205
CYRO\or% 206
CYRP\or% 207
CYRR\or% 208
CYRS\or% 209
CYRT\or% 210
CYRU\or% 211
CYRF\or% 212
CYRH\or% 213
CYRC\or% 214
CYRCH\or% 215
CYRSH\or% 216
CYRSHCH\or% 217
CYRHRDSN\or% 218
CYRERY\or% 219
CYRSFTSN\or% 220
CYREREV\or% 221
CYRYU\or% 222
CYRYA\or% 223
cyra\or% 224
cyrb\or% 225
cyrv\or% 226
cyrg\or% 227
cyrd\or% 228
cyre\or% 229
cyrzh\or% 230
cyrz\or% 231
cyri\or% 232
cyrishrt\or% 233
cyrk\or% 234
cyrl\or% 235
cyrm\or% 236
cyrn\or% 237
cyro\or% 238
cyrp\or% 239
cyrr\or% 240
cyrs\or% 241
cyrt\or% 242
cyru\or% 243
cyrf\or% 244
cyrh\or% 245
cyrc\or% 246
cyrch\or% 247
cyrsh\or% 248
cyrshch\or% 249
cyrhrdsn\or% 250
cyrery\or% 251
cyrsftsn\or% 252
cyrerev\or% 253
cyryu\or% 254
```

```
  cyrya\fi% 255
}
⟨/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.

```
⟨*mtx & t2b⟩
\newcommand\getpsname[1]{%
  \ifcase#1%
  grave\or% 000
  acute\or% 001
  circumflex\or% 002
  tilde\or% 003
  dieresis\or% 004
  hungarumlaut\or% 005
  ring\or% 006
  caron\or% 007
  breve\or% 008
  macron\or% 009
  dotaccent\or% 010
  cedilla\or% 011
  ogonek\or% 012
  CYRpalochka\or% 013
  angleleft\or% 014
  angleright\or% 015
  quotedblleft\or% 016
  quotedblright\or% 017
  cyrflex\or% 018
  dblgrave\or% 019
  cyrbreve\or% 020
  rangedash\or% 021
  cyrdash\or% 022
  compwordmark\or% 023
  perthousandzero\or% 024
  dotlessi\or% 025
  dotlessj\or% 026
  ff\or% 027
  fi\or% 028
  fl\or% 029
  ffi\or% 030
  ffl\or% 031
  visiblespace\or% 032
  exclam\or% 033
  quotedbl\or% 034
  numbersign\or% 035
  dollar\or% 036
  percent\or% 037
  ampersand\or% 038
  quoteright\or% 039
  parenleft\or% 040
  parenright\or% 041
```

```
asterisk\or% 042
plus\or% 043
comma\or% 044
hyphen\or% 045
period\or% 046
slash\or% 047
zero\or% 048
one\or% 049
two\or% 050
three\or% 051
four\or% 052
five\or% 053
six\or% 054
seven\or% 055
eight\or% 056
nine\or% 057
colon\or% 058
semicolon\or% 059
less\or% 060
equal\or% 061
greater\or% 062
question\or% 063
at\or% 064
A\or% 065
B\or% 066
C\or% 067
D\or% 068
E\or% 069
F\or% 070
G\or% 071
H\or% 072
I\or% 073
J\or% 074
K\or% 075
L\or% 076
M\or% 077
N\or% 078
O\or% 079
P\or% 080
Q\or% 081
R\or% 082
S\or% 083
T\or% 084
U\or% 085
V\or% 086
W\or% 087
X\or% 088
Y\or% 089
Z\or% 090
bracketleft\or% 091
backslash\or% 092
bracketright\or% 093
asciicircum\or% 094
underscore\or% 095
```

```
quoteleft\or% 096
a\or% 097
b\or% 098
c\or% 099
d\or% 100
e\or% 101
f\or% 102
g\or% 103
h\or% 104
i\or% 105
j\or% 106
k\or% 107
l\or% 108
m\or% 109
n\or% 110
o\or% 111
p\or% 112
q\or% 113
r\or% 114
s\or% 115
t\or% 116
u\or% 117
v\or% 118
w\or% 119
x\or% 120
y\or% 121
z\or% 122
braceleft\or% 123
bar\or% 124
braceright\or% 125
asciitilde\or% 126
hyphenchar\or% 127
CYRGDSCHCRS\or% 128
CYRGHCRS\or% 129
CYRGDSC\or% 130
CYRGHK\or% 131
CYRSHHA\or% 132
CYRZHDSC\or% 133
CYRDELTA\or% 134
CYRABHDZE\or% 135
CYRLJE\or% 136
CYRKDSC\or% 137
CYRLDSC\or% 138
CYRKHK\or% 139
CYRLHK\or% 140
CYRNDSC\or% 141
CYRNG\or% 142
CYRNHK\or% 143
CYROTLD\or% 144
CYRSACRS\or% 145
CYRUSHRT\or% 146
CYRY\or% 147
CYRHHCRS\or% 148
CYRHDSC\or% 149
```

```
CYRHHK\or% 150
CYRCHLDSC\or% 151
CYRCHRDSC\or% 152
CYRNJE\or% 153
CYRSCHWA\or% 154
CYREPS\or% 155
CYRYO\or% 156
numero\or% 157
currency\or% 158
section\or% 159
cyrgdschcrs\or% 160
cyrghcrs\or% 161
cyrgdsc\or% 162
cyrghk\or% 163
cyrshha\or% 164
cyrzhdsc\or% 165
cyrdelta\or% 166
cyrabhdze\or% 167
cyrlje\or% 168
cyrkdsc\or% 169
cyrldsc\or% 170
cyrkhk\or% 171
cyrlhk\or% 172
cyrndsc\or% 173
cyrng\or% 174
cyrnhk\or% 175
cyrotld\or% 176
cyrsacrs\or% 177
cyrushrt\or% 178
cyry\or% 179
cyrhhcrs\or% 180
cyrhdsc\or% 181
cyrhhk\or% 182
cyrchldsc\or% 183
cyrchrdsc\or% 184
cyrnje\or% 185
cyrschwa\or% 186
cyreps\or% 187
cyryo\or% 188
quotedblbase\or% 189
guillemotleft\or% 190
guillemotright\or% 191
CYRA\or% 192
CYRB\or% 193
CYRV\or% 194
CYRG\or% 195
CYRD\or% 196
CYRE\or% 197
CYRZH\or% 198
CYRZ\or% 199
CYRI\or% 200
CYRISHRT\or% 201
CYRK\or% 202
CYRL\or% 203
```

```
  CYRM\or% 204
  CYRN\or% 205
  CYRO\or% 206
  CYRP\or% 207
  CYRR\or% 208
  CYRS\or% 209
  CYRT\or% 210
  CYRU\or% 211
  CYRF\or% 212
  CYRH\or% 213
  CYRC\or% 214
  CYRCH\or% 215
  CYRSH\or% 216
  CYRSHCH\or% 217
  CYRHRDSN\or% 218
  CYRERY\or% 219
  CYRSFTSN\or% 220
  CYREREV\or% 221
  CYRYU\or% 222
  CYRYA\or% 223
  cyra\or% 224
  cyrb\or% 225
  cyrv\or% 226
  cyrg\or% 227
  cyrd\or% 228
  cyre\or% 229
  cyrzh\or% 230
  cyrz\or% 231
  cyri\or% 232
  cyrishrt\or% 233
  cyrk\or% 234
  cyrl\or% 235
  cyrm\or% 236
  cyrn\or% 237
  cyro\or% 238
  cyrp\or% 239
  cyrr\or% 240
  cyrs\or% 241
  cyrt\or% 242
  cyru\or% 243
  cyrf\or% 244
  cyrh\or% 245
  cyrc\or% 246
  cyrch\or% 247
  cyrsh\or% 248
  cyrshch\or% 249
  cyrhrdsn\or% 250
  cyrery\or% 251
  cyrsftsn\or% 252
  cyrerev\or% 253
  cyryu\or% 254
  cyrya\fi% 255
}
⟨/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.

```
⟨∗mtx & ly1⟩
\newcommand\getpsname[1]{%
  \ifcase#1%
  .notdef.000\or % 000
  Euro\or % 001
  .notdef.002\or % 002
  .notdef.003\or % 003
  fraction\or % 004
  dotaccent\or % 005
  hungarumlaut\or % 006
  ogonek\or % 007
  fl\or % 008
  .notdef.009\or % 009
  cwm\or % 010
  ff\or % 011
  fi\or % 012
  .notdef.013\or % 013
  ffi\or % 014
  ffl\or % 015
  dotlessi\or % 016
  dotlessj\or % 017
  grave\or % 018
  acute\or % 019
  caron\or % 020
  breve\or % 021
  macron\or % 022
  ring\or % 023
  cedilla\or % 024
  germandbls\or % 025
  ae\or % 026
  oe\or % 027
  oslash\or % 028
  AE\or % 029
  OE\or % 030
  Oslash\or % 031
  space\or % 032
  exclam\or % 033
  quotedbl\or % 034
  numbersign\or % 035
  dollar\or % 036
  percent\or % 037
  ampersand\or % 038
  quoteright\or % 039
  parenleft\or % 040
  parenright\or % 041
  asterisk\or % 042
  plus\or % 043
  comma\or % 044
  hyphen\or % 045
```

```
period\or % 046
slash\or % 047
zero\or % 048
one\or % 049
two\or % 050
three\or % 051
four\or % 052
five\or % 053
six\or % 054
seven\or % 055
eight\or % 056
nine\or % 057
colon\or % 058
semicolon\or % 059
less\or % 060
equal\or % 061
greater\or % 062
question\or % 063
at\or % 064
A\or % 065
B\or % 066
C\or % 067
D\or % 068
E\or % 069
F\or % 070
G\or % 071
H\or % 072
I\or % 073
J\or % 074
K\or % 075
L\or % 076
M\or % 077
N\or % 078
O\or % 079
P\or % 080
Q\or % 081
R\or % 082
S\or % 083
T\or % 084
U\or % 085
V\or % 086
W\or % 087
X\or % 088
Y\or % 089
Z\or % 090
bracketleft\or % 091
backslash\or % 092
bracketright\or % 093
circumflex\or % 094
underscore\or % 095
quoteleft\or % 096
a\or % 097
b\or % 098
c\or % 099
```

```
d\or % 100
e\or % 101
f\or % 102
g\or % 103
h\or % 104
i\or % 105
j\or % 106
k\or % 107
l\or % 108
m\or % 109
n\or % 110
o\or % 111
p\or % 112
q\or % 113
r\or % 114
s\or % 115
t\or % 116
u\or % 117
v\or % 118
w\or % 119
x\or % 120
y\or % 121
z\or % 122
braceleft\or % 123
bar\or % 124
braceright\or % 125
tilde\or % 126
dieresis\or % 127
Lslash\or % 128
quotesingle\or % 129
quotesinglbase\or % 130
florin\or % 131
quotedblbase\or % 132
ellipsis\or % 133
dagger\or % 134
daggerdbl\or % 135
circumflex\or % 136
perthousand\or % 137
Scaron\or % 138
guilsinglleft\or % 139
OE\or % 140
Zcaron\or % 141
asciicircum\or % 142
minus\or % 143
lslash\or % 144
quoteleft\or % 145
quoteright\or % 146
quotedblleft\or % 147
quotedblright\or % 148
bullet\or % 149
endash\or % 150
emdash\or % 151
tilde\or % 152
trademark\or % 153
```

```
scaron\or % 154
guilsinglright\or % 155
oe\or % 156
zcaron\or % 157
asciitilde\or % 158
Ydieresis\or % 159
nbspace\or % 160
exclamdown\or % 161
cent\or % 162
sterling\or % 163
currency\or % 164
yen\or % 165
brokenbar\or % 166
section\or % 167
dieresis\or % 168
copyright\or % 169
ordfeminine\or % 170
guillemotleft\or % 171
logicalnot\or % 172
sfthyphen\or % 173
registered\or % 174
macron\or % 175
degree\or % 176
plusminus\or % 177
twosuperior\or % 178
threesuperior\or % 179
acute\or % 180
mu\or % 181
paragraph\or % 182
periodcentered\or % 183
cedilla\or % 184
onesuperior\or % 185
ordmasculine\or % 186
guillemotright\or % 187
onequarter\or % 188
onehalf\or % 189
threequarters\or % 190
questiondown\or % 191
Agrave\or % 192
Aacute\or % 193
Acircumflex\or % 194
Atilde\or % 195
Adieresis\or % 196
Aring\or % 197
AE\or % 198
Ccedilla\or % 199
Egrave\or % 200
Eacute\or % 201
Ecircumflex\or % 202
Edieresis\or % 203
Igrave\or % 204
Iacute\or % 205
Icircumflex\or % 206
Idieresis\or % 207
```

```
  Eth\or % 208
  Ntilde\or % 209
  Ograve\or % 210
  Oacute\or % 211
  Ocircumflex\or % 212
  Otilde\or % 213
  Odieresis\or % 214
  multiply\or % 215
  Oslash\or % 216
  Ugrave\or % 217
  Uacute\or % 218
  Ucircumflex\or % 219
  Udieresis\or % 220
  Yacute\or % 221
  Thorn\or % 222
  germandbls\or % 223
  agrave\or % 224
  aacute\or % 225
  acircumflex\or % 226
  atilde\or % 227
  adieresis\or % 228
  aring\or % 229
  ae\or % 230
  ccedilla\or % 231
  egrave\or % 232
  eacute\or % 233
  ecircumflex\or % 234
  edieresis\or % 235
  igrave\or % 236
  iacute\or % 237
  icircumflex\or % 238
  idieresis\or % 239
  eth\or % 240
  ntilde\or % 241
  ograve\or % 242
  oacute\or % 243
  ocircumflex\or % 244
  otilde\or % 245
  odieresis\or % 246
  divide\or % 247
  oslash\or % 248
  ugrave\or % 249
  uacute\or % 250
  ucircumflex\or % 251
  udieresis\or % 252
  yacute\or % 253
  thorn\or % 254
  ydieresis\fi % 255
}
⟨/mtx & ly1⟩
```

6.4 Templates

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

6.4.1 T1 encoding

```
⟨*template & t1⟩
\listfiles
%% Replace the 'XXX' in the next line by the 3- or 4-character long
%% abbreviation for your font.
\documentclass[family=XXX]{kerntest}

%% Replace the settings by these you want to test.
\kernsetup{encoding=T1,series=m,shape=n,example=hello}
\kernsetup{size=14.40pt,baselineskip=16.5pt,papersize=a4paper}

%% The next line can be used to add a name suffix to the output |mtx| file.
%% \kernsetup{extraname=normal}

%% Set encoding parameters.
%% Set ligaturing: 1=all, 0=all, -1=some, -2=none
%% \encodingsetup{ligaturing=1}

%% If you are using a font with different design sizes and if you want
%% to test one design size scaled to another one, you may input a
%% modified fd file for your font. To generate this new fd file, just
%% copy the original one, rename it, and modify the entries for the
%% font shapes to use the design size you want to test.
%% \input{t1XXX-1200.fd}

%% The following lines show some possible glyph classes. You should
%% add all classes you need.
\newglyphclass{right}{A}{A,Aogonek}
\newglyphclass{right}{Abreve}{Abreve,Aacute,Acircumflex,%
  Atilde,Adieresis}
\newglyphclass{right}{Aring}{Agrave,Aring}

\newglyphclass{left}{A}{A,Aogonek}
\newglyphclass{left}{Abreve}{Abreve,Agrave,Acircumflex,%
  Atilde,Adieresis}
\newglyphclass{left}{Aring}{Aacute,Aring}

\newglyphclass{right}{C}{C,Cacute,Ccaron,Ccedilla}
\newglyphclass{left}{C}{C,Cacute,Ccaron,Ccedilla}

\newglyphclass{right}{D}{D,Dcaron,Eth}

\newglyphclass{right}{E}{E,Ecaron,Eogonek,Egrave,Eacute,Ecircumflex,%
  Edieresis,AE,OE}

\newglyphclass{right}{G}{G,Gbreve}
\newglyphclass{left}{G}{G,Gbreve}

\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,N,Nacute,Ncaron,%
  Ntilde,P,R,Racute,Rcaron,Ng,Thorn}

```

```
\newglyphclass{right}{H}{H,I,Idotaccent,Igrave,Iacute,%
  Icircumflex,Idieresis,IJ,J,N,Nacute,Ncaron,Ntilde}

\newglyphclass{right}{O}{O,Ohungarumlaut,Ograve,Oacute,Ocircumflex,%
  Otilde,Odieresis,Oslash}
\newglyphclass{left}{O}{O,Ohungarumlaut,Ograve,Oacute,Ocircumflex,%
  Otilde,Odieresis,OE,Oslash}

\newglyphclass{right}{R}{R,Racute,Rcaron}

\newglyphclass{right}{S}{S,Sacute,Scaron,Scedilla,SS,dollar}
\newglyphclass{left}{S}{S,Sacute,Scaron,Scedilla,SS,dollar}

\newglyphclass{right}{T}{T,Tcaron,Tcedilla}
\newglyphclass{left}{T}{T,Tcaron,Tcedilla}

\newglyphclass{right}{U}{U,Uhungarumlaut,Uring,Ugrave,Uacute,%
  Ucircumflex,Udieresis}
\newglyphclass{left}{U}{U,Uhungarumlaut,Uring,Ugrave,Uacute,%
  Ucircumflex,Udieresis}

\newglyphclass{right}{Y}{Y,Yacute,Ydieresis}%
\newglyphclass{left}{Y}{Y,Yacute,Ydieresis}%

\newglyphclass{left}{Z}{Z,Zacute,Zcaron,Zdotaccent}
\newglyphclass{right}{Z}{Z,Zacute,Zcaron,Zdotaccent}

\newglyphclass{left}{a}{a,aogonek,ae}
\newglyphclass{left}{abreve}{abreve,agrave,acircumflex,atilde,adieresis,aring}
\newglyphclass{right}{a}{a,aogonek}
\newglyphclass{right}{abreve}{abreve,aacute,acircumflex,atilde,adieresis,aring}

\newglyphclass{left}{c}{c,ccedilla}
\newglyphclass{right}{c}{c,ccedilla}

\newglyphclass{left}{d}{d,dbar,dcaron}

\newglyphclass{left}{e}{e,eogonek}
\newglyphclass{left}{egrave}{ecaron,ecircumflex}
\newglyphclass{right}{e}{e,eogonek,ae,oe}
\newglyphclass{right}{egrave}{ecaron,ecircumflex}

\newglyphclass{left}{f}{f,ff,fi,fl,ffi,ffl}
\newglyphclass{right}{f}{f,ff}

\newglyphclass{right}{i}{i,fi,ffi,dotlessi}
\newglyphclass{left}{i}{i,ij,dotlessi}

\newglyphclass{right}{l}{fl,ffl,l,lacute}
\newglyphclass{left}{l}{l,lacute,lcaron}

\newglyphclass{right}{m}{m,n}
\newglyphclass{left}{m}{m,n}
\newglyphclass{right}{nacute}{nacute,ncaron}
```

```
\newglyphclass{left}{nacute}{nacute,ncaron}

\newglyphclass{right}{ograve}{ograve,ocircumflex}
\newglyphclass{right}{ohungarumlaut}{ohungarumlaut,oacute,otilde,%
  odieresis}
\newglyphclass{left}{oacute}{oacute,ocircumflex,oe}
\newglyphclass{left}{ohungarumlaut}{ohungarumlaut,ograve,otilde,%
  odieresis}

\newglyphclass{left}{t}{t,tcaron,tcedilla}


\begin{document}

%% This table of characters is sorted by similar glyphs, not by the
%% encoding.
%% Replace `{LLL}' and `{RRR}' in columns 1 resp. 5 by these glyphs
%% that shall be tested.
%% Good pairs to use for LLL and RRR test are 016, 017 (English
%% quotation marks `` and ''); 018, 016 (German quotation marks ,, and
%% ``); 019, 020 (French Guillemets << and >>); 020, 019 (French
%% Guillemets in German notation >> and <<).
%% You may also take the Postscript names for all glyphs.
\begin{kerntable}
  \testkern{LLL}{-}{000}{-}{RRR} \\ % grave
  \testkern{LLL}{-}{001}{-}{RRR} \\ % acute
  \testkern{LLL}{-}{002}{-}{RRR} \\ % circumflex
  \testkern{LLL}{-}{003}{-}{RRR} \\ % tilde
  \testkern{LLL}{-}{004}{-}{RRR} \\ % dieresis
  \testkern{LLL}{-}{005}{-}{RRR} \\ % hungarumlaut
  \testkern{LLL}{-}{006}{-}{RRR} \\ % ring
  \testkern{LLL}{-}{007}{-}{RRR} \\ % caron
  \testkern{LLL}{-}{008}{-}{RRR} \\ % breve
  \testkern{LLL}{-}{009}{-}{RRR} \\ % macron
  \testkern{LLL}{-}{010}{-}{RRR} \\ % dotaccent
  \testkern{LLL}{-}{011}{-}{RRR} \\ % cedilla
  \testkern{LLL}{-}{012}{-}{RRR} \\ % ogonek
  \testkern{LLL}{-}{126}{-}{RRR} \\ % asciitilde
  \testkern{LLL}{-}{094}{-}{RRR} \\ % asciicircum
  \testkern{LLL}{-}{042}{-}{RRR} \\ % asterisk
  \testkern{LLL}{-}{023}{-}{RRR} \\ % compwordmark
  \testkern{LLL}{-}{032}{-}{RRR} \\ % visiblespace
  \testkern{LLL}{-}{014}{-}{RRR} \\ % guilsinglleft
  \testkern{LLL}{-}{015}{-}{RRR} \\ % guilsinglright
  \testkern{LLL}{-}{019}{-}{RRR} \\ % guillemotleft
  \testkern{LLL}{-}{020}{-}{RRR} \\ % guillemotright
  \testkern{LLL}{-}{096}{-}{RRR} \\ % quoteleft
  \testkern{LLL}{-}{039}{-}{RRR} \\ % quoteright
  \testkern{LLL}{-}{013}{-}{RRR} \\ % quotesinglbase
  \testkern{LLL}{-}{016}{-}{RRR} \\ % quotedblleft
  \testkern{LLL}{-}{017}{-}{RRR} \\ % quotedblright
  \testkern{LLL}{-}{018}{-}{RRR} \\ % quotedblbase
  \testkern{LLL}{-}{034}{-}{RRR} \\ % quotedbl
  \testkern{LLL}{-}{046}{-}{RRR} \\ % period
```

```
\testkern{LLL}{-}{044}{-}{RRR} \\ % comma
\testkern{LLL}{-}{058}{-}{RRR} \\ % colon
\testkern{LLL}{-}{059}{-}{RRR} \\ % semicolon
\testkern{LLL}{-}{033}{-}{RRR} \\ % exclam
\testkern{LLL}{-}{063}{-}{RRR} \\ % question
\testkern{LLL}{-}{189}{-}{RRR} \\ % exclamdown
\testkern{LLL}{-}{190}{-}{RRR} \\ % questiondown
\testkern{LLL}{-}{045}{-}{RRR} \\ % hyphen
\testkern{LLL}{-}{127}{-}{RRR} \\ % hyphenchar
\testkern{LLL}{-}{021}{-}{RRR} \\ % rangedash
\testkern{LLL}{-}{022}{-}{RRR} \\ % punctdash
\testkern{LLL}{-}{095}{-}{RRR} \\ % underscore
\testkern{LLL}{-}{043}{-}{RRR} \\ % plus
\testkern{LLL}{-}{061}{-}{RRR} \\ % equal
\testkern{LLL}{-}{060}{-}{RRR} \\ % less
\testkern{LLL}{-}{062}{-}{RRR} \\ % greater
\testkern{LLL}{-}{047}{-}{RRR} \\ % slash
\testkern{LLL}{-}{092}{-}{RRR} \\ % backslash
\testkern{LLL}{-}{040}{-}{RRR} \\ % parenleft
\testkern{LLL}{-}{041}{-}{RRR} \\ % parenright
\testkern{LLL}{-}{091}{-}{RRR} \\ % bracketleft
\testkern{LLL}{-}{093}{-}{RRR} \\ % bracketright
\testkern{LLL}{-}{123}{-}{RRR} \\ % braceleft
\testkern{LLL}{-}{125}{-}{RRR} \\ % braceright
\testkern{LLL}{-}{124}{-}{RRR} \\ % bar
\testkern{LLL}{-}{035}{-}{RRR} \\ % numbersign
\testkern{LLL}{-}{037}{-}{RRR} \\ % percent
\testkern{LLL}{-}{024}{-}{RRR} \\ % perthousandzero
\testkern{LLL}{-}{038}{-}{RRR} \\ % ampersand
\testkern{LLL}{-}{159}{-}{RRR} \\ % section
\testkern{LLL}{-}{064}{-}{RRR} \\ % at
\testkern{LLL}{-}{191}{-}{RRR} \\ % sterling
\testkern{LLL}{-}{048}{-}{RRR} \\ % zero
\testkern{LLL}{-}{049}{-}{RRR} \\ % one
\testkern{LLL}{-}{050}{-}{RRR} \\ % two
\testkern{LLL}{-}{051}{-}{RRR} \\ % three
\testkern{LLL}{-}{052}{-}{RRR} \\ % four
\testkern{LLL}{-}{053}{-}{RRR} \\ % five
\testkern{LLL}{-}{054}{-}{RRR} \\ % six
\testkern{LLL}{-}{055}{-}{RRR} \\ % seven
\testkern{LLL}{-}{056}{-}{RRR} \\ % eight
\testkern{LLL}{-}{057}{-}{RRR} \\ % nine
\testkern{LLL}{-}{065}{-}{RRR} \\ % A
\testkern{LLL}{-}{128}{-}{RRR} \\ % Abreve
\testkern{LLL}{-}{129}{-}{RRR} \\ % Aogonek
\testkern{LLL}{-}{192}{-}{RRR} \\ % Agrave
\testkern{LLL}{-}{193}{-}{RRR} \\ % Aacute
\testkern{LLL}{-}{194}{-}{RRR} \\ % Acircumflex
\testkern{LLL}{-}{195}{-}{RRR} \\ % Atilde
\testkern{LLL}{-}{196}{-}{RRR} \\ % Adieresis
\testkern{LLL}{-}{197}{-}{RRR} \\ % Aring
\testkern{LLL}{-}{198}{-}{RRR} \\ % AE
\testkern{LLL}{-}{066}{-}{RRR} \\ % B
\testkern{LLL}{-}{067}{-}{RRR} \\ % C
```

```
\testkern{LLL}{-}{130}{-}{RRR} \\ % Cacute
\testkern{LLL}{-}{131}{-}{RRR} \\ % Ccaron
\testkern{LLL}{-}{199}{-}{RRR} \\ % Ccedilla
\testkern{LLL}{-}{068}{-}{RRR} \\ % D
\testkern{LLL}{-}{132}{-}{RRR} \\ % Dcaron
\testkern{LLL}{-}{208}{-}{RRR} \\ % Eth
\testkern{LLL}{-}{069}{-}{RRR} \\ % E
\testkern{LLL}{-}{133}{-}{RRR} \\ % Ecaron
\testkern{LLL}{-}{134}{-}{RRR} \\ % Eogonek
\testkern{LLL}{-}{200}{-}{RRR} \\ % Egrave
\testkern{LLL}{-}{201}{-}{RRR} \\ % Eacute
\testkern{LLL}{-}{202}{-}{RRR} \\ % Ecircumflex
\testkern{LLL}{-}{203}{-}{RRR} \\ % Edieresis
\testkern{LLL}{-}{070}{-}{RRR} \\ % F
\testkern{LLL}{-}{071}{-}{RRR} \\ % G
\testkern{LLL}{-}{135}{-}{RRR} \\ % Gbreve
\testkern{LLL}{-}{072}{-}{RRR} \\ % H
\testkern{LLL}{-}{073}{-}{RRR} \\ % I
\testkern{LLL}{-}{157}{-}{RRR} \\ % Idotaccent
\testkern{LLL}{-}{204}{-}{RRR} \\ % Igrave
\testkern{LLL}{-}{205}{-}{RRR} \\ % Iacute
\testkern{LLL}{-}{206}{-}{RRR} \\ % Icircumflex
\testkern{LLL}{-}{207}{-}{RRR} \\ % Idieresis
\testkern{LLL}{-}{156}{-}{RRR} \\ % IJ
\testkern{LLL}{-}{074}{-}{RRR} \\ % J
\testkern{LLL}{-}{075}{-}{RRR} \\ % K
\testkern{LLL}{-}{076}{-}{RRR} \\ % L
\testkern{LLL}{-}{136}{-}{RRR} \\ % Lacute
\testkern{LLL}{-}{137}{-}{RRR} \\ % Lcaron
\testkern{LLL}{-}{138}{-}{RRR} \\ % Lslash
\testkern{LLL}{-}{077}{-}{RRR} \\ % M
\testkern{LLL}{-}{078}{-}{RRR} \\ % N
\testkern{LLL}{-}{139}{-}{RRR} \\ % Nacute
\testkern{LLL}{-}{140}{-}{RRR} \\ % Ncaron
\testkern{LLL}{-}{209}{-}{RRR} \\ % Ntilde
\testkern{LLL}{-}{079}{-}{RRR} \\ % O
\testkern{LLL}{-}{142}{-}{RRR} \\ % Ohungarumlaut
\testkern{LLL}{-}{210}{-}{RRR} \\ % Ograve
\testkern{LLL}{-}{211}{-}{RRR} \\ % Oacute
\testkern{LLL}{-}{212}{-}{RRR} \\ % Ocircumflex
\testkern{LLL}{-}{213}{-}{RRR} \\ % Otilde
\testkern{LLL}{-}{214}{-}{RRR} \\ % Odieresis
\testkern{LLL}{-}{215}{-}{RRR} \\ % OE
\testkern{LLL}{-}{216}{-}{RRR} \\ % Oslash
\testkern{LLL}{-}{080}{-}{RRR} \\ % P
\testkern{LLL}{-}{081}{-}{RRR} \\ % Q
\testkern{LLL}{-}{082}{-}{RRR} \\ % R
\testkern{LLL}{-}{143}{-}{RRR} \\ % Racute
\testkern{LLL}{-}{144}{-}{RRR} \\ % Rcaron
\testkern{LLL}{-}{083}{-}{RRR} \\ % S
\testkern{LLL}{-}{145}{-}{RRR} \\ % Sacute
\testkern{LLL}{-}{146}{-}{RRR} \\ % Scaron
\testkern{LLL}{-}{147}{-}{RRR} \\ % Scedilla
\testkern{LLL}{-}{223}{-}{RRR} \\ % SS
```

```
\testkern{LLL}{-}{036}{-}{RRR} \\ % dollar
\testkern{LLL}{-}{084}{-}{RRR} \\ % T
\testkern{LLL}{-}{148}{-}{RRR} \\ % Tcaron
\testkern{LLL}{-}{149}{-}{RRR} \\ % Tcedilla
\testkern{LLL}{-}{085}{-}{RRR} \\ % U
\testkern{LLL}{-}{150}{-}{RRR} \\ % Uhungarumlaut
\testkern{LLL}{-}{151}{-}{RRR} \\ % Uring
\testkern{LLL}{-}{217}{-}{RRR} \\ % Ugrave
\testkern{LLL}{-}{218}{-}{RRR} \\ % Uacute
\testkern{LLL}{-}{219}{-}{RRR} \\ % Ucircumflex
\testkern{LLL}{-}{220}{-}{RRR} \\ % Udieresis
\testkern{LLL}{-}{086}{-}{RRR} \\ % V
\testkern{LLL}{-}{087}{-}{RRR} \\ % W
\testkern{LLL}{-}{088}{-}{RRR} \\ % X
\testkern{LLL}{-}{089}{-}{RRR} \\ % Y
\testkern{LLL}{-}{152}{-}{RRR} \\ % Ydieresis
\testkern{LLL}{-}{221}{-}{RRR} \\ % Yacute
\testkern{LLL}{-}{090}{-}{RRR} \\ % Z
\testkern{LLL}{-}{153}{-}{RRR} \\ % Zacute
\testkern{LLL}{-}{154}{-}{RRR} \\ % Zcaron
\testkern{LLL}{-}{155}{-}{RRR} \\ % Zdotaccent
\testkern{LLL}{-}{141}{-}{RRR} \\ % Ng
\testkern{LLL}{-}{222}{-}{RRR} \\ % Thorn
\testkern{LLL}{-}{097}{-}{RRR} \\ % a
\testkern{LLL}{-}{160}{-}{RRR} \\ % abreve
\testkern{LLL}{-}{161}{-}{RRR} \\ % aogonek
\testkern{LLL}{-}{224}{-}{RRR} \\ % agrave
\testkern{LLL}{-}{225}{-}{RRR} \\ % aacute
\testkern{LLL}{-}{226}{-}{RRR} \\ % acircumflex
\testkern{LLL}{-}{227}{-}{RRR} \\ % atilde
\testkern{LLL}{-}{228}{-}{RRR} \\ % adieresis
\testkern{LLL}{-}{229}{-}{RRR} \\ % aring
\testkern{LLL}{-}{230}{-}{RRR} \\ % ae
\testkern{LLL}{-}{098}{-}{RRR} \\ % b
\testkern{LLL}{-}{099}{-}{RRR} \\ % c
\testkern{LLL}{-}{162}{-}{RRR} \\ % cacute
\testkern{LLL}{-}{163}{-}{RRR} \\ % ccaron
\testkern{LLL}{-}{231}{-}{RRR} \\ % ccedilla
\testkern{LLL}{-}{100}{-}{RRR} \\ % d
\testkern{LLL}{-}{158}{-}{RRR} \\ % dbar
\testkern{LLL}{-}{164}{-}{RRR} \\ % dcaron
\testkern{LLL}{-}{101}{-}{RRR} \\ % e
\testkern{LLL}{-}{165}{-}{RRR} \\ % ecaron
\testkern{LLL}{-}{166}{-}{RRR} \\ % eogonek
\testkern{LLL}{-}{232}{-}{RRR} \\ % egrave
\testkern{LLL}{-}{233}{-}{RRR} \\ % eacute
\testkern{LLL}{-}{234}{-}{RRR} \\ % ecircumflex
\testkern{LLL}{-}{235}{-}{RRR} \\ % edieresis
\testkern{LLL}{-}{102}{-}{RRR} \\ % f
\testkern{LLL}{-}{027}{-}{RRR} \\ % ff
\testkern{LLL}{-}{028}{-}{RRR} \\ % fi
\testkern{LLL}{-}{029}{-}{RRR} \\ % fl
\testkern{LLL}{-}{030}{-}{RRR} \\ % ffi
\testkern{LLL}{-}{031}{-}{RRR} \\ % ffl
```

```
\testkern{LLL}{-}{103}{-}{RRR} \\ % g
\testkern{LLL}{-}{167}{-}{RRR} \\ % gbreve
\testkern{LLL}{-}{104}{-}{RRR} \\ % h
\testkern{LLL}{-}{105}{-}{RRR} \\ % i
\testkern{LLL}{-}{025}{-}{RRR} \\ % dotlessi
\testkern{LLL}{-}{236}{-}{RRR} \\ % igrave
\testkern{LLL}{-}{237}{-}{RRR} \\ % iacute
\testkern{LLL}{-}{238}{-}{RRR} \\ % icircumflex
\testkern{LLL}{-}{239}{-}{RRR} \\ % idieresis
\testkern{LLL}{-}{188}{-}{RRR} \\ % ij
\testkern{LLL}{-}{106}{-}{RRR} \\ % j
\testkern{LLL}{-}{026}{-}{RRR} \\ % dotlessj
\testkern{LLL}{-}{107}{-}{RRR} \\ % k
\testkern{LLL}{-}{108}{-}{RRR} \\ % l
\testkern{LLL}{-}{168}{-}{RRR} \\ % lacute
\testkern{LLL}{-}{169}{-}{RRR} \\ % lcaron
\testkern{LLL}{-}{170}{-}{RRR} \\ % lslash
\testkern{LLL}{-}{109}{-}{RRR} \\ % m
\testkern{LLL}{-}{110}{-}{RRR} \\ % n
\testkern{LLL}{-}{171}{-}{RRR} \\ % nacute
\testkern{LLL}{-}{172}{-}{RRR} \\ % ncaron
\testkern{LLL}{-}{241}{-}{RRR} \\ % ntilde
\testkern{LLL}{-}{111}{-}{RRR} \\ % o
\testkern{LLL}{-}{174}{-}{RRR} \\ % ohungarumlaut
\testkern{LLL}{-}{242}{-}{RRR} \\ % ograve
\testkern{LLL}{-}{243}{-}{RRR} \\ % oacute
\testkern{LLL}{-}{244}{-}{RRR} \\ % ocircumflex
\testkern{LLL}{-}{245}{-}{RRR} \\ % otilde
\testkern{LLL}{-}{246}{-}{RRR} \\ % odieresis
\testkern{LLL}{-}{247}{-}{RRR} \\ % oe
\testkern{LLL}{-}{248}{-}{RRR} \\ % oslash
\testkern{LLL}{-}{112}{-}{RRR} \\ % p
\testkern{LLL}{-}{113}{-}{RRR} \\ % q
\testkern{LLL}{-}{114}{-}{RRR} \\ % r
\testkern{LLL}{-}{175}{-}{RRR} \\ % racute
\testkern{LLL}{-}{176}{-}{RRR} \\ % rcaron
\testkern{LLL}{-}{115}{-}{RRR} \\ % s
\testkern{LLL}{-}{177}{-}{RRR} \\ % sacute
\testkern{LLL}{-}{178}{-}{RRR} \\ % scaron
\testkern{LLL}{-}{179}{-}{RRR} \\ % scedilla
\testkern{LLL}{-}{116}{-}{RRR} \\ % t
\testkern{LLL}{-}{180}{-}{RRR} \\ % tcaron
\testkern{LLL}{-}{181}{-}{RRR} \\ % tcedilla
\testkern{LLL}{-}{117}{-}{RRR} \\ % u
\testkern{LLL}{-}{182}{-}{RRR} \\ % uhungarumlaut
\testkern{LLL}{-}{183}{-}{RRR} \\ % uring
\testkern{LLL}{-}{249}{-}{RRR} \\ % ugrave
\testkern{LLL}{-}{250}{-}{RRR} \\ % uacute
\testkern{LLL}{-}{251}{-}{RRR} \\ % ucircumflex
\testkern{LLL}{-}{252}{-}{RRR} \\ % udieresis
\testkern{LLL}{-}{118}{-}{RRR} \\ % v
\testkern{LLL}{-}{119}{-}{RRR} \\ % w
\testkern{LLL}{-}{120}{-}{RRR} \\ % x
\testkern{LLL}{-}{121}{-}{RRR} \\ % y
```

```
  \testkern{LLL}{-}{184}{-}{RRR} \\ % ydieresis
  \testkern{LLL}{-}{253}{-}{RRR} \\ % yacute
  \testkern{LLL}{-}{122}{-}{RRR} \\ % z
  \testkern{LLL}{-}{185}{-}{RRR} \\ % zacute
  \testkern{LLL}{-}{186}{-}{RRR} \\ % zcaron
  \testkern{LLL}{-}{187}{-}{RRR} \\ % zdotaccent
  \testkern{LLL}{-}{240}{-}{RRR} \\ % eth
  \testkern{LLL}{-}{173}{-}{RRR} \\ % ng
  \testkern{LLL}{-}{254}{-}{RRR} \\ % thorn
  \testkern{LLL}{-}{255}{-}{RRR} \\ % germandbls
\end{kerntable}

\end{document}
```
⟨/template & t1⟩

6.4.2 TS1 encoding

```
⟨*template & ts1⟩
\listfiles
%% Replace the 'XXX' in the next line by the 3- or 4-character long
%% abbreviation for your font.
\documentclass[family=XXX]{kerntest}

%% Replace the settings by these you want to test.
\kernsetup{encoding=TS1,series=m,shape=n,example=hello}
\kernsetup{size=14.40pt,baselineskip=16.5pt,papersize=a4paper}

%% The next line can be used to add a name suffix to the output |mtx| file.
%% \kernsetup{extraname=normal}

%% If you are using a font with different design sizes and if you want
%% to test one design size scaled to another one, you may input a
%% modified fd file for your font. To generate this new fd file, just
%% copy the original one, rename it, and modify the entries for the
%% font shapes to use the design size you want to test.
%% \input{t1XXX-1200.fd}

\begin{document}

%% This table of characters is sorted by similar glyphs, not by the
%% encoding.
%% Replace '{LLL}' and '{RRR}' in columns 1 resp. 5 by these glyphs
%% that shall be tested.
\begin{kerntable}
  \testkern{LLL}{-}{000}{-}{RRR} \\ % capitalgrave
  \testkern{LLL}{-}{001}{-}{RRR} \\ % capitalacute
  \testkern{LLL}{-}{002}{-}{RRR} \\ % capitalcircumflex
  \testkern{LLL}{-}{003}{-}{RRR} \\ % capitaltilde
  \testkern{LLL}{-}{004}{-}{RRR} \\ % capitaldieresis
  \testkern{LLL}{-}{005}{-}{RRR} \\ % capitalhungarumlaut
  \testkern{LLL}{-}{006}{-}{RRR} \\ % capitalring
  \testkern{LLL}{-}{007}{-}{RRR} \\ % capitalcaron
  \testkern{LLL}{-}{008}{-}{RRR} \\ % capitalbreve
  \testkern{LLL}{-}{009}{-}{RRR} \\ % capitalmacron
  \testkern{LLL}{-}{010}{-}{RRR} \\ % capitaldotaccent
```

```
\testkern{LLL}{-}{011}{-}{RRR} \\ % cedilla
\testkern{LLL}{-}{012}{-}{RRR} \\ % ogonek
\testkern{LLL}{-}{013}{-}{RRR} \\ % quotesinglbase
\testkern{LLL}{-}{014}{-}{RRR} \\
\testkern{LLL}{-}{015}{-}{RRR} \\
\testkern{LLL}{-}{016}{-}{RRR} \\
\testkern{LLL}{-}{017}{-}{RRR} \\
\testkern{LLL}{-}{018}{-}{RRR} \\ % quotedblbase
\testkern{LLL}{-}{019}{-}{RRR} \\
\testkern{LLL}{-}{020}{-}{RRR} \\
\testkern{LLL}{-}{021}{-}{RRR} \\ % twelveudash
\testkern{LLL}{-}{022}{-}{RRR} \\ % threequartersemdash
\testkern{LLL}{-}{023}{-}{RRR} \\ % capitalcompwordmark
\testkern{LLL}{-}{024}{-}{RRR} \\ % arrowleft
\testkern{LLL}{-}{025}{-}{RRR} \\ % arrowright
\testkern{LLL}{-}{026}{-}{RRR} \\ % tieaccentlowercase
\testkern{LLL}{-}{027}{-}{RRR} \\ % tieaccentcapital
\testkern{LLL}{-}{028}{-}{RRR} \\ % newtieaccentlowercase
\testkern{LLL}{-}{029}{-}{RRR} \\ % newtieaccentcapital
\testkern{LLL}{-}{030}{-}{RRR} \\ % ascendercompwordmark
\testkern{LLL}{-}{031}{-}{RRR} \\ % blank
\testkern{LLL}{-}{032}{-}{RRR} \\
\testkern{LLL}{-}{033}{-}{RRR} \\
\testkern{LLL}{-}{034}{-}{RRR} \\
\testkern{LLL}{-}{035}{-}{RRR} \\
\testkern{LLL}{-}{036}{-}{RRR} \\ % dollar
\testkern{LLL}{-}{037}{-}{RRR} \\
\testkern{LLL}{-}{038}{-}{RRR} \\
\testkern{LLL}{-}{039}{-}{RRR} \\ % quotesingle
\testkern{LLL}{-}{040}{-}{RRR} \\
\testkern{LLL}{-}{041}{-}{RRR} \\
\testkern{LLL}{-}{042}{-}{RRR} \\ % asteriskcentered
\testkern{LLL}{-}{043}{-}{RRR} \\
\testkern{LLL}{-}{044}{-}{RRR} \\ % comma
\testkern{LLL}{-}{045}{-}{RRR} \\ % hyphendbl
\testkern{LLL}{-}{046}{-}{RRR} \\ % period
\testkern{LLL}{-}{047}{-}{RRR} \\ % fraction
\testkern{LLL}{-}{048}{-}{RRR} \\ % zerooldstyle
\testkern{LLL}{-}{049}{-}{RRR} \\ % oneoldstyle
\testkern{LLL}{-}{050}{-}{RRR} \\ % twooldstyle
\testkern{LLL}{-}{051}{-}{RRR} \\ % threeoldstyle
\testkern{LLL}{-}{052}{-}{RRR} \\ % fouroldstyle
\testkern{LLL}{-}{053}{-}{RRR} \\ % fiveoldstyle
\testkern{LLL}{-}{054}{-}{RRR} \\ % sixoldstyle
\testkern{LLL}{-}{055}{-}{RRR} \\ % sevenoldstyle
\testkern{LLL}{-}{056}{-}{RRR} \\ % eightoldstyle
\testkern{LLL}{-}{057}{-}{RRR} \\ % nineoldstyle
\testkern{LLL}{-}{058}{-}{RRR} \\
\testkern{LLL}{-}{059}{-}{RRR} \\
\testkern{LLL}{-}{060}{-}{RRR} \\ % angbracketleft
\testkern{LLL}{-}{061}{-}{RRR} \\ % minus
\testkern{LLL}{-}{062}{-}{RRR} \\ % angbracketright
\testkern{LLL}{-}{063}{-}{RRR} \\
\testkern{LLL}{-}{064}{-}{RRR} \\
```

```
\testkern{LLL}{-}{065}{-}{RRR} \\
\testkern{LLL}{-}{066}{-}{RRR} \\
\testkern{LLL}{-}{067}{-}{RRR} \\
\testkern{LLL}{-}{068}{-}{RRR} \\
\testkern{LLL}{-}{069}{-}{RRR} \\
\testkern{LLL}{-}{070}{-}{RRR} \\
\testkern{LLL}{-}{071}{-}{RRR} \\
\testkern{LLL}{-}{072}{-}{RRR} \\
\testkern{LLL}{-}{073}{-}{RRR} \\
\testkern{LLL}{-}{074}{-}{RRR} \\
\testkern{LLL}{-}{075}{-}{RRR} \\
\testkern{LLL}{-}{076}{-}{RRR} \\
\testkern{LLL}{-}{077}{-}{RRR} \\ % Omegainv
\testkern{LLL}{-}{078}{-}{RRR} \\
\testkern{LLL}{-}{079}{-}{RRR} \\ % bigcircle
\testkern{LLL}{-}{080}{-}{RRR} \\
\testkern{LLL}{-}{081}{-}{RRR} \\
\testkern{LLL}{-}{082}{-}{RRR} \\
\testkern{LLL}{-}{083}{-}{RRR} \\
\testkern{LLL}{-}{084}{-}{RRR} \\
\testkern{LLL}{-}{085}{-}{RRR} \\
\testkern{LLL}{-}{086}{-}{RRR} \\
\testkern{LLL}{-}{087}{-}{RRR} \\ % Omega
\testkern{LLL}{-}{088}{-}{RRR} \\
\testkern{LLL}{-}{089}{-}{RRR} \\
\testkern{LLL}{-}{090}{-}{RRR} \\
\testkern{LLL}{-}{091}{-}{RRR} \\ % openbracketleft
\testkern{LLL}{-}{092}{-}{RRR} \\
\testkern{LLL}{-}{093}{-}{RRR} \\ % openbracketright
\testkern{LLL}{-}{094}{-}{RRR} \\ % arrowup
\testkern{LLL}{-}{095}{-}{RRR} \\ % arrowdown
\testkern{LLL}{-}{096}{-}{RRR} \\ % asciigrave
\testkern{LLL}{-}{097}{-}{RRR} \\
\testkern{LLL}{-}{098}{-}{RRR} \\ % born
\testkern{LLL}{-}{099}{-}{RRR} \\ % divorced
\testkern{LLL}{-}{100}{-}{RRR} \\ % died
\testkern{LLL}{-}{101}{-}{RRR} \\
\testkern{LLL}{-}{102}{-}{RRR} \\
\testkern{LLL}{-}{103}{-}{RRR} \\
\testkern{LLL}{-}{104}{-}{RRR} \\
\testkern{LLL}{-}{105}{-}{RRR} \\
\testkern{LLL}{-}{106}{-}{RRR} \\
\testkern{LLL}{-}{107}{-}{RRR} \\
\testkern{LLL}{-}{108}{-}{RRR} \\ % leaf
\testkern{LLL}{-}{109}{-}{RRR} \\ % married
\testkern{LLL}{-}{110}{-}{RRR} \\ % musicalnote
\testkern{LLL}{-}{111}{-}{RRR} \\
\testkern{LLL}{-}{112}{-}{RRR} \\
\testkern{LLL}{-}{113}{-}{RRR} \\
\testkern{LLL}{-}{114}{-}{RRR} \\
\testkern{LLL}{-}{115}{-}{RRR} \\
\testkern{LLL}{-}{116}{-}{RRR} \\
\testkern{LLL}{-}{117}{-}{RRR} \\
\testkern{LLL}{-}{118}{-}{RRR} \\
```

```
\testkern{LLL}{-}{119}{-}{RRR} \\
\testkern{LLL}{-}{120}{-}{RRR} \\
\testkern{LLL}{-}{121}{-}{RRR} \\
\testkern{LLL}{-}{122}{-}{RRR} \\
\testkern{LLL}{-}{123}{-}{RRR} \\
\testkern{LLL}{-}{124}{-}{RRR} \\
\testkern{LLL}{-}{125}{-}{RRR} \\
\testkern{LLL}{-}{126}{-}{RRR} \\ % tildelow
\testkern{LLL}{-}{127}{-}{RRR} \\ % hyphendblchar
\testkern{LLL}{-}{128}{-}{RRR} \\ % asciibreve
\testkern{LLL}{-}{129}{-}{RRR} \\ % asciicaron
\testkern{LLL}{-}{130}{-}{RRR} \\ % asciiacutedbl
\testkern{LLL}{-}{131}{-}{RRR} \\ % asciigravedbl
\testkern{LLL}{-}{132}{-}{RRR} \\ % dagger
\testkern{LLL}{-}{133}{-}{RRR} \\ % daggerdbl
\testkern{LLL}{-}{134}{-}{RRR} \\ % bardbl
\testkern{LLL}{-}{135}{-}{RRR} \\ % perthousand
\testkern{LLL}{-}{136}{-}{RRR} \\ % bullet
\testkern{LLL}{-}{137}{-}{RRR} \\ % centigrade
\testkern{LLL}{-}{138}{-}{RRR} \\ % dollaroldstyle
\testkern{LLL}{-}{139}{-}{RRR} \\ % centoldstyle
\testkern{LLL}{-}{140}{-}{RRR} \\ % florin
\testkern{LLL}{-}{141}{-}{RRR} \\ % colonmonetary
\testkern{LLL}{-}{142}{-}{RRR} \\ % won
\testkern{LLL}{-}{143}{-}{RRR} \\ % naira
\testkern{LLL}{-}{144}{-}{RRR} \\ % guarani
\testkern{LLL}{-}{145}{-}{RRR} \\ % peso
\testkern{LLL}{-}{146}{-}{RRR} \\ % lira
\testkern{LLL}{-}{147}{-}{RRR} \\ % recipe
\testkern{LLL}{-}{148}{-}{RRR} \\ % interrobang
\testkern{LLL}{-}{149}{-}{RRR} \\ % interrobangdown
\testkern{LLL}{-}{150}{-}{RRR} \\ % dong
\testkern{LLL}{-}{151}{-}{RRR} \\ % trademark
\testkern{LLL}{-}{152}{-}{RRR} \\ % pertenthousand
\testkern{LLL}{-}{153}{-}{RRR} \\ % pilcrow
\testkern{LLL}{-}{154}{-}{RRR} \\ % baht
\testkern{LLL}{-}{155}{-}{RRR} \\ % numero
\testkern{LLL}{-}{156}{-}{RRR} \\ % discount
\testkern{LLL}{-}{157}{-}{RRR} \\ % estimated
\testkern{LLL}{-}{158}{-}{RRR} \\ % openbullet
\testkern{LLL}{-}{159}{-}{RRR} \\ % servicemark
\testkern{LLL}{-}{160}{-}{RRR} \\ % quillbracketleft
\testkern{LLL}{-}{161}{-}{RRR} \\ % quillbracketright
\testkern{LLL}{-}{162}{-}{RRR} \\ % cent
\testkern{LLL}{-}{163}{-}{RRR} \\ % sterling
\testkern{LLL}{-}{164}{-}{RRR} \\ % currency
\testkern{LLL}{-}{165}{-}{RRR} \\ % yen
\testkern{LLL}{-}{166}{-}{RRR} \\ % brokenbar
\testkern{LLL}{-}{167}{-}{RRR} \\ % section
\testkern{LLL}{-}{168}{-}{RRR} \\ % asciidieresis
\testkern{LLL}{-}{169}{-}{RRR} \\ % copyright
\testkern{LLL}{-}{170}{-}{RRR} \\ % ordfeminine
\testkern{LLL}{-}{171}{-}{RRR} \\ % copyleft
\testkern{LLL}{-}{172}{-}{RRR} \\ % logicalnot
```

```
\testkern{LLL}{-}{173}{-}{RRR} \\ % circledP
\testkern{LLL}{-}{174}{-}{RRR} \\ % registered
\testkern{LLL}{-}{175}{-}{RRR} \\ % asciimacron
\testkern{LLL}{-}{176}{-}{RRR} \\ % degree
\testkern{LLL}{-}{177}{-}{RRR} \\ % plusminus
\testkern{LLL}{-}{178}{-}{RRR} \\ % twosuperior
\testkern{LLL}{-}{179}{-}{RRR} \\ % threesuperior
\testkern{LLL}{-}{180}{-}{RRR} \\ % asciiacute
\testkern{LLL}{-}{181}{-}{RRR} \\ % mu
\testkern{LLL}{-}{182}{-}{RRR} \\ % paragraph
\testkern{LLL}{-}{183}{-}{RRR} \\ % periodcentered
\testkern{LLL}{-}{184}{-}{RRR} \\ % referencemark
\testkern{LLL}{-}{185}{-}{RRR} \\ % onesuperior
\testkern{LLL}{-}{186}{-}{RRR} \\ % ordmasculine
\testkern{LLL}{-}{187}{-}{RRR} \\ % radical
\testkern{LLL}{-}{188}{-}{RRR} \\ % onequarter
\testkern{LLL}{-}{189}{-}{RRR} \\ % onehalf
\testkern{LLL}{-}{190}{-}{RRR} \\ % threequarters
\testkern{LLL}{-}{191}{-}{RRR} \\ % euro
\testkern{LLL}{-}{192}{-}{RRR} \\
\testkern{LLL}{-}{193}{-}{RRR} \\
\testkern{LLL}{-}{194}{-}{RRR} \\
\testkern{LLL}{-}{195}{-}{RRR} \\
\testkern{LLL}{-}{196}{-}{RRR} \\
\testkern{LLL}{-}{197}{-}{RRR} \\
\testkern{LLL}{-}{198}{-}{RRR} \\
\testkern{LLL}{-}{199}{-}{RRR} \\
\testkern{LLL}{-}{200}{-}{RRR} \\
\testkern{LLL}{-}{201}{-}{RRR} \\
\testkern{LLL}{-}{202}{-}{RRR} \\
\testkern{LLL}{-}{203}{-}{RRR} \\
\testkern{LLL}{-}{204}{-}{RRR} \\
\testkern{LLL}{-}{205}{-}{RRR} \\
\testkern{LLL}{-}{206}{-}{RRR} \\
\testkern{LLL}{-}{207}{-}{RRR} \\
\testkern{LLL}{-}{208}{-}{RRR} \\
\testkern{LLL}{-}{209}{-}{RRR} \\
\testkern{LLL}{-}{210}{-}{RRR} \\
\testkern{LLL}{-}{211}{-}{RRR} \\
\testkern{LLL}{-}{212}{-}{RRR} \\
\testkern{LLL}{-}{213}{-}{RRR} \\
\testkern{LLL}{-}{214}{-}{RRR} \\ % multiply
\testkern{LLL}{-}{215}{-}{RRR} \\
\testkern{LLL}{-}{216}{-}{RRR} \\
\testkern{LLL}{-}{217}{-}{RRR} \\
\testkern{LLL}{-}{218}{-}{RRR} \\
\testkern{LLL}{-}{219}{-}{RRR} \\
\testkern{LLL}{-}{220}{-}{RRR} \\
\testkern{LLL}{-}{221}{-}{RRR} \\
\testkern{LLL}{-}{222}{-}{RRR} \\
\testkern{LLL}{-}{223}{-}{RRR} \\
\testkern{LLL}{-}{224}{-}{RRR} \\
\testkern{LLL}{-}{225}{-}{RRR} \\
\testkern{LLL}{-}{226}{-}{RRR} \\
```

```
  \testkern{LLL}{-}{227}{-}{RRR} \\
  \testkern{LLL}{-}{228}{-}{RRR} \\
  \testkern{LLL}{-}{229}{-}{RRR} \\
  \testkern{LLL}{-}{230}{-}{RRR} \\
  \testkern{LLL}{-}{231}{-}{RRR} \\
  \testkern{LLL}{-}{232}{-}{RRR} \\
  \testkern{LLL}{-}{233}{-}{RRR} \\
  \testkern{LLL}{-}{234}{-}{RRR} \\
  \testkern{LLL}{-}{235}{-}{RRR} \\
  \testkern{LLL}{-}{236}{-}{RRR} \\
  \testkern{LLL}{-}{237}{-}{RRR} \\
  \testkern{LLL}{-}{238}{-}{RRR} \\
  \testkern{LLL}{-}{239}{-}{RRR} \\
  \testkern{LLL}{-}{240}{-}{RRR} \\
  \testkern{LLL}{-}{241}{-}{RRR} \\
  \testkern{LLL}{-}{242}{-}{RRR} \\
  \testkern{LLL}{-}{243}{-}{RRR} \\
  \testkern{LLL}{-}{244}{-}{RRR} \\
  \testkern{LLL}{-}{245}{-}{RRR} \\
  \testkern{LLL}{-}{246}{-}{RRR} \\ % divide
  \testkern{LLL}{-}{247}{-}{RRR} \\
  \testkern{LLL}{-}{248}{-}{RRR} \\
  \testkern{LLL}{-}{249}{-}{RRR} \\
  \testkern{LLL}{-}{250}{-}{RRR} \\
  \testkern{LLL}{-}{251}{-}{RRR} \\
  \testkern{LLL}{-}{252}{-}{RRR} \\
  \testkern{LLL}{-}{253}{-}{RRR} \\
  \testkern{LLL}{-}{254}{-}{RRR} \\
  \testkern{LLL}{-}{255}{-}{RRR} \\
\end{kerntable}

\end{document}
⟨/template & ts1⟩
```

6.4.3 OT1 encoding

```
⟨*template & ot1⟩
\listfiles
%% Replace the 'XXX' in the next line by the 3- or 4-character long
%% abbreviation for your font.
\documentclass[family=XXX]{kerntest}

%% Replace the settings by these you want to test.
\kernsetup{encoding=OT1,series=m,shape=n,example=hello}
\kernsetup{size=14.40pt,baselineskip=16.5pt,papersize=a4paper}

%% The next line can be used to add a name suffix to the output mtx file.
%% \kernsetup{extraname=normal}

%% Set encoding parameters.
%% Set ligaturing: 2=all, 1=some, 0=none
%% \encodingsetup{ligaturing=2}
%% Normal: dollar, Italic: sterling
%% \encodingsetup{italicizing=false}

```

```
%% If you are using a font with different design sizes and if you want
%% to test one design size scaled to another one, you may input a
%% modified fd file for your font. To generate this new fd file, just
%% copy the original one, rename it, and modify the entries for the
%% font shapes to use the design size you want to test.
%% \input{ot1XXX-1200.fd}

%% The following lines show some possible glyph classes. You should
%% add all classes you need.
\newglyphclass{right}{E}{E,AE,OE}

\newglyphclass{left}{H}{B,D,F,H,I,J,K,L,N,P,R}

\newglyphclass{right}{H}{H,I,J,N}

\newglyphclass{right}{S}{S,dollar}
\newglyphclass{left}{S}{S,dollar}

\newglyphclass{left}{a}{a,ae}
\newglyphclass{right}{e}{e,ae,oe}

\newglyphclass{left}{f}{f,ff,fi,fl,ffi,ffl}
\newglyphclass{right}{f}{f,ff}

\newglyphclass{right}{i}{i,fi,ffi,dotlessi}
\newglyphclass{left}{i}{i,dotlessi}

\newglyphclass{right}{l}{fl,ffl,l}

\newglyphclass{right}{m}{m,n}
\newglyphclass{left}{m}{m,n}

\newglyphclass{left}{o}{o,oe}

\begin{document}

%% This table of characters is sorted by similar glyphs, not by the
%% encoding.
%% Replace `{LLL}' and `{RRR}' in columns 1 resp. 5 by these glyphs
%% that shall be tested.
\begin{kerntable}
  \testkern{LLL}{-}{018}{-}{RRR} \\ % grave
  \testkern{LLL}{-}{019}{-}{RRR} \\ % acute
  \testkern{LLL}{-}{094}{-}{RRR} \\ % circumflex
  \testkern{LLL}{-}{126}{-}{RRR} \\ % tilde
  \testkern{LLL}{-}{127}{-}{RRR} \\ % dieresis
  \testkern{LLL}{-}{125}{-}{RRR} \\ % hungarumlaut/braceright
  \testkern{LLL}{-}{023}{-}{RRR} \\ % ringfitted
  \testkern{LLL}{-}{020}{-}{RRR} \\ % caron
  \testkern{LLL}{-}{021}{-}{RRR} \\ % breve
  \testkern{LLL}{-}{022}{-}{RRR} \\ % macron
  \testkern{LLL}{-}{095}{-}{RRR} \\ % dotaccent/underscore
  \testkern{LLL}{-}{024}{-}{RRR} \\ % cedilla
  \testkern{LLL}{-}{042}{-}{RRR} \\ % asterisk
```

```
\testkern{LLL}{-}{032}{-}{RRR} \\ % lslashslash/visiblespace
\testkern{LLL}{-}{096}{-}{RRR} \\ % quoteleft
\testkern{LLL}{-}{039}{-}{RRR} \\ % quoteright
\testkern{LLL}{-}{092}{-}{RRR} \\ % quotedblleft
\testkern{LLL}{-}{034}{-}{RRR} \\ % quotedblright/quotedbl
\testkern{LLL}{-}{046}{-}{RRR} \\ % period
\testkern{LLL}{-}{044}{-}{RRR} \\ % comma
\testkern{LLL}{-}{058}{-}{RRR} \\ % colon
\testkern{LLL}{-}{059}{-}{RRR} \\ % semicolon
\testkern{LLL}{-}{033}{-}{RRR} \\ % exclam
\testkern{LLL}{-}{063}{-}{RRR} \\ % question
\testkern{LLL}{-}{060}{-}{RRR} \\ % exclamdown/less
\testkern{LLL}{-}{062}{-}{RRR} \\ % questiondown/greater
\testkern{LLL}{-}{045}{-}{RRR} \\ % hyphen
\testkern{LLL}{-}{123}{-}{RRR} \\ % rangedash/braceleft
\testkern{LLL}{-}{124}{-}{RRR} \\ % punctdash/bar
\testkern{LLL}{-}{043}{-}{RRR} \\ % plus
\testkern{LLL}{-}{061}{-}{RRR} \\ % equal
\testkern{LLL}{-}{047}{-}{RRR} \\ % slash
\testkern{LLL}{-}{040}{-}{RRR} \\ % parenleft
\testkern{LLL}{-}{041}{-}{RRR} \\ % parenright
\testkern{LLL}{-}{091}{-}{RRR} \\ % bracketleft
\testkern{LLL}{-}{093}{-}{RRR} \\ % bracketright
\testkern{LLL}{-}{035}{-}{RRR} \\ % numbersign
\testkern{LLL}{-}{037}{-}{RRR} \\ % percent
\testkern{LLL}{-}{038}{-}{RRR} \\ % ampersand
\testkern{LLL}{-}{064}{-}{RRR} \\ % at
\testkern{LLL}{-}{048}{-}{RRR} \\ % zero
\testkern{LLL}{-}{049}{-}{RRR} \\ % one
\testkern{LLL}{-}{050}{-}{RRR} \\ % two
\testkern{LLL}{-}{051}{-}{RRR} \\ % three
\testkern{LLL}{-}{052}{-}{RRR} \\ % four
\testkern{LLL}{-}{053}{-}{RRR} \\ % five
\testkern{LLL}{-}{054}{-}{RRR} \\ % six
\testkern{LLL}{-}{055}{-}{RRR} \\ % seven
\testkern{LLL}{-}{056}{-}{RRR} \\ % eight
\testkern{LLL}{-}{057}{-}{RRR} \\ % nine
\testkern{LLL}{-}{065}{-}{RRR} \\ % A
\testkern{LLL}{-}{029}{-}{RRR} \\ % AE
\testkern{LLL}{-}{066}{-}{RRR} \\ % B
\testkern{LLL}{-}{067}{-}{RRR} \\ % C
\testkern{LLL}{-}{068}{-}{RRR} \\ % D
\testkern{LLL}{-}{069}{-}{RRR} \\ % E
\testkern{LLL}{-}{070}{-}{RRR} \\ % F
\testkern{LLL}{-}{071}{-}{RRR} \\ % G
\testkern{LLL}{-}{072}{-}{RRR} \\ % H
\testkern{LLL}{-}{073}{-}{RRR} \\ % I
\testkern{LLL}{-}{074}{-}{RRR} \\ % J
\testkern{LLL}{-}{075}{-}{RRR} \\ % K
\testkern{LLL}{-}{076}{-}{RRR} \\ % L
\testkern{LLL}{-}{138}{-}{RRR} \\ % Lslash
\testkern{LLL}{-}{077}{-}{RRR} \\ % M
\testkern{LLL}{-}{078}{-}{RRR} \\ % N
\testkern{LLL}{-}{079}{-}{RRR} \\ % O
```

```
\testkern{LLL}{-}{030}{-}{RRR} \\ % OE
\testkern{LLL}{-}{031}{-}{RRR} \\ % Oslash
\testkern{LLL}{-}{080}{-}{RRR} \\ % P
\testkern{LLL}{-}{081}{-}{RRR} \\ % Q
\testkern{LLL}{-}{082}{-}{RRR} \\ % R
\testkern{LLL}{-}{083}{-}{RRR} \\ % S
\testkern{LLL}{-}{036}{-}{RRR} \\ % dollar/sterling
\testkern{LLL}{-}{084}{-}{RRR} \\ % T
\testkern{LLL}{-}{085}{-}{RRR} \\ % U
\testkern{LLL}{-}{086}{-}{RRR} \\ % V
\testkern{LLL}{-}{087}{-}{RRR} \\ % W
\testkern{LLL}{-}{088}{-}{RRR} \\ % X
\testkern{LLL}{-}{089}{-}{RRR} \\ % Y
\testkern{LLL}{-}{090}{-}{RRR} \\ % Z
\testkern{LLL}{-}{097}{-}{RRR} \\ % a
\testkern{LLL}{-}{026}{-}{RRR} \\ % ae
\testkern{LLL}{-}{098}{-}{RRR} \\ % b
\testkern{LLL}{-}{099}{-}{RRR} \\ % c
\testkern{LLL}{-}{100}{-}{RRR} \\ % d
\testkern{LLL}{-}{101}{-}{RRR} \\ % e
\testkern{LLL}{-}{102}{-}{RRR} \\ % f
\testkern{LLL}{-}{011}{-}{RRR} \\ % ff/arrowup
\testkern{LLL}{-}{012}{-}{RRR} \\ % fi/arrowdown
\testkern{LLL}{-}{013}{-}{RRR} \\ % fl/quotesingle
\testkern{LLL}{-}{014}{-}{RRR} \\ % ffi/exclamdown
\testkern{LLL}{-}{015}{-}{RRR} \\ % ffl/questiondown
\testkern{LLL}{-}{103}{-}{RRR} \\ % g
\testkern{LLL}{-}{104}{-}{RRR} \\ % h
\testkern{LLL}{-}{105}{-}{RRR} \\ % i
\testkern{LLL}{-}{016}{-}{RRR} \\ % dotlessi
\testkern{LLL}{-}{025}{-}{RRR} \\ % germandbls
\testkern{LLL}{-}{106}{-}{RRR} \\ % j
\testkern{LLL}{-}{017}{-}{RRR} \\ % dotlessj
\testkern{LLL}{-}{107}{-}{RRR} \\ % k
\testkern{LLL}{-}{108}{-}{RRR} \\ % l
\testkern{LLL}{-}{170}{-}{RRR} \\ % lslash
\testkern{LLL}{-}{109}{-}{RRR} \\ % m
\testkern{LLL}{-}{110}{-}{RRR} \\ % n
\testkern{LLL}{-}{111}{-}{RRR} \\ % o
\testkern{LLL}{-}{027}{-}{RRR} \\ % oe
\testkern{LLL}{-}{028}{-}{RRR} \\ % oslash
\testkern{LLL}{-}{112}{-}{RRR} \\ % p
\testkern{LLL}{-}{113}{-}{RRR} \\ % q
\testkern{LLL}{-}{114}{-}{RRR} \\ % r
\testkern{LLL}{-}{115}{-}{RRR} \\ % s
\testkern{LLL}{-}{116}{-}{RRR} \\ % t
\testkern{LLL}{-}{117}{-}{RRR} \\ % u
\testkern{LLL}{-}{118}{-}{RRR} \\ % v
\testkern{LLL}{-}{119}{-}{RRR} \\ % w
\testkern{LLL}{-}{120}{-}{RRR} \\ % x
\testkern{LLL}{-}{121}{-}{RRR} \\ % y
\testkern{LLL}{-}{122}{-}{RRR} \\ % z
\testkern{LLL}{-}{000}{-}{RRR} \\ % Gamma
\testkern{LLL}{-}{001}{-}{RRR} \\ % Delta
```

```
  \testkern{LLL}{-}{002}{-}{RRR} \\ % Theta
  \testkern{LLL}{-}{003}{-}{RRR} \\ % Lambda
  \testkern{LLL}{-}{004}{-}{RRR} \\ % Xi
  \testkern{LLL}{-}{005}{-}{RRR} \\ % Pi
  \testkern{LLL}{-}{006}{-}{RRR} \\ % Sigma
  \testkern{LLL}{-}{007}{-}{RRR} \\ % Upsilon
  \testkern{LLL}{-}{008}{-}{RRR} \\ % Phi
  \testkern{LLL}{-}{009}{-}{RRR} \\ % Psi
  \testkern{LLL}{-}{010}{-}{RRR} \\ % Omega
\iffalse
  \testkern{LLL}{-}{128}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{129}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{130}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{131}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{132}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{133}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{134}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{135}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{136}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{137}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{139}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{140}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{141}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{142}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{143}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{144}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{145}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{146}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{147}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{148}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{149}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{150}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{151}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{152}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{153}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{154}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{155}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{156}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{157}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{158}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{159}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{160}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{161}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{162}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{163}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{164}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{165}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{166}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{167}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{168}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{169}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{171}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{172}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{173}{-}{RRR} \\ % .notdef.
```

```
\testkern{LLL}{-}{174}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{175}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{176}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{177}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{178}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{179}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{180}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{181}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{182}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{183}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{184}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{185}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{186}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{187}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{188}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{189}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{190}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{191}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{192}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{193}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{194}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{195}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{196}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{197}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{198}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{199}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{200}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{201}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{202}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{203}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{204}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{205}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{206}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{207}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{208}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{209}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{210}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{211}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{212}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{213}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{214}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{215}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{216}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{217}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{218}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{219}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{220}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{221}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{222}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{223}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{224}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{225}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{226}{-}{RRR} \\ % .notdef.
\testkern{LLL}{-}{227}{-}{RRR} \\ % .notdef.
```

```
  \testkern{LLL}{-}{228}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{229}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{230}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{231}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{232}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{233}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{234}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{235}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{236}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{237}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{238}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{239}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{240}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{241}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{242}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{243}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{244}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{245}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{246}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{247}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{248}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{249}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{250}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{251}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{252}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{253}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{254}{-}{RRR} \\ % .notdef.
  \testkern{LLL}{-}{255}{-}{RRR} \\ % .notdef.
\fi
\end{kerntable}

\end{document}
⟨/template & ot1⟩
```

That's it.

Change History

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.

D

E

F

G