

# ltoutenc.dtx

Johannes Braams      David Carlisle      Alan Jeffrey  
Frank Mittelbach      Chris Rowley      Rainer Schöpf

2017/04/05

This file is maintained by the L<sup>A</sup>T<sub>E</sub>X Project team.  
Bug reports can be opened (category `latex`) at  
<https://latex-project.org/bugs.html>.

## 1 Font encodings

This section of the kernel contains commands for declaring encoding-specific commands, such as accents. It also contains the code for some of the encoding files, including `omlenc.def`, `omsenc.def`, `t1enc.def` and `ot1enc.def` files, which define the OLM, OMS, T1 and OT1 encodings, and the `fontenc` package for selecting encodings.

The `fontenc` package has options for encodings, of which the last option is the default encoding. For example, to use the OT2, OT3 and T1 encodings, with T1 as the default, you say:

```
\usepackage[OT2,OT3,T1]{fontenc}
```

The standard kernel set-up loads font encoding files and selects an encoding as follows.

```
\input {omlenc.def}  
\input {t1enc.def}  
\input {ot1enc.def}  
\input {omsenc.def}  
\fontencoding{OT1}
```

Note that the files in the standard `inputenc` package depend on this behaviour of the kernel.

The syntax for declaring encoding-specific commands is:

```
\DeclareTextCommand{<command>}{<encoding>}  
[<number>][<default>]{<commands>}
```

This command is like `\newcommand`, except that it defines a command which is specific to one encoding. The resulting command is always robust, even if its definition is fragile. For example, the definition of `\l` in the OT1 encoding is:

```
\DeclareTextCommand{\l}{OT1}{\@xxxii l}
```



`\DeclareTextCommand` takes the same optional arguments as `\newcommand`.

```
\ProvideTextCommand{<command>}{<encoding>}
                        [<number>] [<default>]{<commands>}
```

This acts like `\DeclareTextCommand`, but does nothing if the command is already defined.

```
\DeclareTextSymbol{<command>}{<encoding>}{<slot>}
```

This command defines a text symbol, with a particular slot in that encoding. The commands:

```
\DeclareTextSymbol{\ss}{OT1}{25}
\DeclareTextCommand{\ss}{OT1}{\char25 }
```

have the same effect, but the `\DeclareTextSymbol` is faster.

```
\DeclareTextAccent{<command>}{<encoding>}{<slot>}
```

This command declares a text accent. The commands:

```
\DeclareTextAccent{"}{OT1}{127}
\DeclareTextCommand{"}{OT1}{\add@accent {127}}
```

have the same effect.

```
\DeclareTextComposite{<command>}
                        {<encoding>}{<argument>}{<slot>}
```

This command declares a composite letter, for example in the T1 encoding `\'a` is slot 225, which is declared by:

```
\DeclareTextComposite{\'}{T1}{a}{225}
```

The *command* will normally have been declared with `\DeclareTextAccent`, or as a one-argument `\DeclareTextCommand`.

`\DeclareTextComposite` is the most common example of using the more general declaration `\DeclareTextCompositeCommand`, which can define a composite to be an arbitrary piece of text.

```
\DeclareTextCompositeCommand{<command>}
                        {<encoding>}{<argument>}{<text>}
```

For example, in the OT1 encoding Å has a hand-crafted definition this is declared as follows

```
\DeclareTextCompositeCommand{\r}{OT1}{A}
{\leavevmode\setbox\z@\hbox{!}\dimen@ht\z@\advance\dimen@-1ex%
 \rlap{\raise.67\dimen@\hbox{\char23}}A}
```



The *command* will normally have been declared with `\DeclareTextAccent`, or as a one-argument `\DeclareTextCommand`.

The commands defined using the above declarations can be used in two ways. Normally they are used by just calling the command in the appropriate encoding, for example `\ss`. However, sometimes you may wish to use a command in an encoding where it is not defined. If the command has no arguments, then you can use it in another encoding by calling `\UseTextSymbol`:

```
\UseTextSymbol{<encoding>}{<command>}
```

For example, `\UseTextSymbol{OT1}{\ss}` has the same effect as:

```
{\fontencoding{OT1}\selectfont\ss}
```

If the command has one argument then you can use it in another encoding by calling `\UseTextAccent`:

```
\UseTextAccent{<encoding>}{<command>}{<text>}
```

For example, if the current encoding is OT2 then `\UseTextAccent{OT1}{\'}{a}` has the same effect as:

```
{\fontencoding{OT1}\selectfont\'{\fontencoding{OT2}\selectfont a}}
```

You can also declare a default definition for a text command, which will be used if the current encoding has no appropriate definition. Such use will also set the definition for this command in the current encoding to equal this default definition; this makes subsequent uses of the command much faster.

```
\DeclareTextCommandDefault{<command>}{<definition>}
```

For example, the default definition of the command `\textonequarter` (which produces the fraction  $\frac{1}{4}$ ) could be built using math mode:

```
\DeclareTextCommandDefault{\textonequarter}{\ensuremath {\frac{1}{4}}}
```

There is a matching `\Provide` command which will not override an existing default definition:

```
\ProvideTextCommandDefault{<command>}{<definition>}
```

The most common use for these commands is to use symbols from other encodings, so there are some optimizations provided:

```
\DeclareTextSymbolDefault{<command>}{<encoding>}
\DeclareTextAccentDefault{<command>}{<encoding>}
```

are short for:

```
\DeclareTextCommandDefault{<command>}
      {\UseTextSymbol{<encoding>}{<command>}}
\DeclareTextCommandDefault[1]{<command>}
      {\UseTextAccent{<encoding>}{<command>}{#1}}
```



For example, to make OT1 the default encoding for `\ss` and `\'` you say:

```
\DeclareTextSymbolDefault{\ss}{OT1}
\DeclareTextAccentDefault{\'}{OT1}
```

Note that you can use these commands on any zero- or one-argument commands declared with `\DeclareText*` or `\ProvideText*`, not just those defined using `\DeclareTextSymbol` or `\DeclareTextAccent`.

## 1.1 Removing encoding-specific commands

In some cases encoding definitions are given to provide some limited support since nothing better is available, for example, the definition for `\textdollar` in OT1 is a hack since \$ and £ actually share the same slot in this encoding. Thus if such a glyph becomes available in a different encoding (e.g., TS1) one would like to get rid of the flacky one and make the default definition point to the new encoding. In such a case defining

```
\DeclareTextSymbol{\textdollar}{TS1}{36}
\DeclareTextSymbolDefault{\textdollar}{TS1}
```

is not enough since if typesetting in OT1 L<sup>A</sup>T<sub>E</sub>X will still find the encoding specific-definition for OT1 and therefore ignore the new default. Therefore to ensure that in this case the TS1 version is used we have to remove the OT1 declaration:

```
\UndeclareTextCommand{\textdollar}{OT1}
```

Since the \$ sign is a proper glyph in the T1 encoding there is no point removing its definition and forcing L<sup>A</sup>T<sub>E</sub>X to pick up the TS1 version if typesetting in this encoding. However, assume you want to use the variant dollar sign, i.e., \$ for your dollars. In that case you have to get rid of the T1 declaration as well, e.g., the following would do that for you:

```
\UndeclareTextCommand{\textdollar}{OT1}
\UndeclareTextCommand{\textdollar}{T1}
\DeclareTextCommandDefault{\textdollar}
{\UseTextSymbol{TS1}\textdollaroldstyle}
```

## 1.2 The order of declarations

If an encoding-specific command is defined for more than one encoding, then it will execute fastest in the encoding in which it was defined last since its top-level definition will be set up to execute in that encoding without any overhead.

For this reason the file `fonttext.ltx` currently first loads the definitions for the T1 encoding and then those for the OT1 encoding so that typesetting in OT1 is optimized since that is (still) the default. However, when T1 is explicitly requested (via `\usepackage[T1]{fontenc}`) the top-level definitions are automatically changed to favour T1 since its declarations are reloaded in the process.

For the same reason default declarations should never come last since they are implemented as a special encoding themselves (with the name ?). Specifying



them last would simply mean to make those encoding-specific commands equally inefficient in all encodings. Therefore the `textcomp` package, for example, first sets up all defaults to point to `TS1` and then declares the commands in the `TS1` encoding.

### 1.3 Docstrip modules

This `.dtx` file is be used to generate several related files containing font encoding definitions. The mutually exclusive docstrip options are listed here.

<code>T1</code>	generates <code>t1enc.def</code> for the Cork encoding.
<code>TS1</code>	generates <code>ts1enc.def</code> for the Text Companion encoding.
<code>TS1sty</code>	generates <code>textcomp.sty</code> , package that sets up use of the Text Companion encoding.
<code>OT1</code>	generates <code>ot1enc.def</code> for Knuth's CM encoding.
<code>OMS</code>	generates <code>omsenc.def</code> for Knuth's math symbol encoding.
<code>OML</code>	generates <code>omlenc.def</code> for Knuth's math letters encoding.
<code>OT4</code>	generates <code>ot4enc.def</code> for the Polish extension to the OT1 encoding, created by B. Jackowski and M. Ryćko for use with the Polish version of Computer Modern and Computer Concrete.
<code>TU</code>	generates <code>tuenc.def</code> for Unicode font encoding.
<code>package</code>	generates <code>fontenc.sty</code> for selecting encodings.
<code>2ekernel</code>	for the kernel commands.

### 1.4 Definitions for the kernel

#### 1.4.1 Declaration commands

This section contains definitions for commands such as accents which depend on the current encoding. These commands will usually be kept in `.def` files, for example `ot1enc.def` contains the definitions for the `OT1` encoding.

```

1 <*2ekernel>
2 \message{font encodings,}
   Far too many macros in one block here!

\DeclareTextCommand If you say:
\ProvideTextCommand
\DeclareTextSymbol   \DeclareTextCommand{\foo}{T1}...
   \@dec@text@cmd then \foo is defined to be \T1-cmd \foo \T1\foo, where \T1\foo is one control
   \chardef@text@cmd sequence, not two! We then call \newcommand to define \T1\foo.
   \@changed@cmd
   \@changed@x
\TextSymbolUnavailable
   \inmathwarn
3 \def\DeclareTextCommand{%
4   \@dec@text@cmd\newcommand}
5 \def\ProvideTextCommand{%
6   \@dec@text@cmd\providecommand}
7 \def\@dec@text@cmd#1#2#3{%
8   \expandafter\def\expandafter#2%
```



```

9      \expandafter{%
10         \csname#3-cmd\expandafter\endcsname
11         \expandafter#2%
12         \csname#3\string#2\endcsname
13     }%
14     \let\@ifdefinable\@rc@ifdefinable
15     \expandafter#1\csname#3\string#2\endcsname}

```

This command was introduced to fix a major bug in `\@dec@text@cmd` without changing that command itself. This was thought to be necessary because it is defined in more than one package. (Perhaps the more serious bug is to put complex low-level commands like this in packages?)

The problem it solves is that whereas both `\newcommand` and `\providecommand` (used just above) both handle the resetting of `\@ifdefinable` (following its disabling in `\@dec@text@cmd`), the primitive `\chardef` neither needs the disabling, nor does the resetting.

```

16 \def\chardef@text@cmd{%
17     \let\@ifdefinable\@ifdefinable
18     \chardef
19 }
20 \def\DeclareTextSymbol#1#2#3{%
21     \@dec@text@cmd\chardef@text@cmd#1{#2}#3\relax
22 }

```

The declarations are only available before `\begin{document}`.

```

23 \@onlypreamble\DeclareTextCommand
24 \@onlypreamble\DeclareTextSymbol

```

The sneaky bit in all this is what `\T1-cmd \foo \T1\foo` does. There are five possibilities, depending on the current values of `\protect`, `\cf@encoding` and `\ifmmode`:

- If `\protect` is `\@typeset@protect` and `\cf@encoding` is `T1`, then we execute `\T1\foo`. This should be the normal behaviour, and is optimized for speed.
- If `\protect` is `\@typeset@protect`, `\cf@encoding` is (say) `OT1`, and `\OT1\foo` is defined, then we execute `\OT1\foo`.
- If `\protect` is `\@typeset@protect`, `\cf@encoding` is (say) `OT1`, we're in text mode, and `\OT1\foo` is undefined, then we define `\OT1\foo` to be the default value of `\foo`, and execute `\OT1\foo`.
- If `\protect` is `\@typeset@protect`, `\cf@encoding` is (say) `OT1`, we're in math mode, and `\OT1\foo` is undefined, then we execute the default value of `\foo`. (This is necessary so that things like `$X_\copyright$` work properly.)
- If `\protect` is not `\@typeset@protect` then we execute `\noexpand\foo`. For example, if we are writing to a file, then this results in `\foo` being written. If we are in a `\mark`, then `\foo` will be put in the mark—since `\foo` is robust, it will then survive all the things which may happen to it whilst it's a `\mark`.



So after all that, we will either execute the appropriate definition of `\foo` for the current encoding, or we will execute `\noexpand\foo`.

The default value of `\foo` is `\?\foo` if it is defined, and an error message otherwise.

When the encoding is changed from T1 to OT1, `\T1-cmd` is defined to be `\@changed@cmd` and `\OT1-cmd` is defined to be `\@current@cmd`. This means that the test for what the current encoding is can be performed quickly.

```

25 \def\@current@cmd#1{%
26   \ifx\protect\@typeset@protect
27     \@inmathwarn#1%
28   \else
29     \noexpand#1\expandafter\@gobble
30   \fi}

31 \def\@changed@cmd#1#2{%
32   \ifx\protect\@typeset@protect
33     \@inmathwarn#1%
34     \expandafter\ifx\csname\cf@encoding\string#1\endcsname\relax
35       \expandafter\ifx\csname ?\string#1\endcsname\relax
36         \expandafter\def\csname ?\string#1\endcsname{%
37           \TextSymbolUnavailable#1%
38         }%
39       \fi
40       \global\expandafter\let
41         \csname\cf@encoding\string#1\expandafter\endcsname
42         \csname ?\string#1\endcsname
43       \fi
44       \csname\cf@encoding\string#1%
45         \expandafter\endcsname
46     \else
47       \noexpand#1%
48     \fi}

49 \gdef\TextSymbolUnavailable#1{%
50   \@latex@error{%
51     Command \protect#1 unavailable in encoding \cf@encoding%
52   }\@eha}

```

The command `\@inmathwarn` produces a warning message if we are currently in math mode. Note that since this command is used inside text commands, it can't call `\relax` before the `\ifmmode`. This means that it is possible for the warning to fail to be issued at the beginning of a row of an `halign` whose template enters math mode. This is probably a bad feature, but there's not much that can be done about it, since adding a `\relax` would break ligatures and kerning between text symbols.

A more efficient solution would be to make `\@inmathwarn` and `\@inmatherr` equal to `\@empty` and `\relax` by default, and to have `\everymath` reset them to their usual definitions. This is left for future investigation (for example it may break some third party code).



```

53 \def\@inmathwarn#1{%
54   \ifmmode
55     \@latex@warning{Command \protect#1 invalid in math mode}%
56   \fi}

```

`\DeclareTextCommandDefault` These define commands with encoding ?.

`\ProvideTextCommandDefault` Note that `\DeclareTextCommandDefault` can only be used in the preamble, but that the `\Provide` version is allowed in inputenc .def files, so is allowed anywhere.

```

57 \def\DeclareTextCommandDefault#1{%
58   \DeclareTextCommand#1?}

59 \def\ProvideTextCommandDefault#1{%
60   \ProvideTextCommand#1?}

61 \@onlypreamble\DeclareTextCommandDefault
62 %\@onlypreamble\ProvideTextCommandDefault

```

They require `?-cmd` to be initialized as `\@changed@cmd`.

```

63 \expandafter\let\csname?~cmd\endcsname\@changed@cmd

```

`\DeclareTextAccent` This is just a disguise for defining a TeX `\accent` command.

```

64 \def\DeclareTextAccent#1#2#3{%
65   \DeclareTextCommand#1{#2}{\add@accent{#3}}
66 \@onlypreamble\DeclareTextAccent

```

`\add@accent` To save space this code is shared between all text accents that are set using the `\accent` primitive. The argument is pre-set in a box so that any font loading that is needed is already done within the box. This is needed because font-loading involves grouping and that would prevent the accent mechanism from working so that the accent would not be positioned over the argument. Declarations that change the font should be allowed (only low-level ones are at present) inside the argument of an accent command, but not size changes, as they involve `\setbox` operations which also inhibit the mechanism of the `\accent` primitive.

Note that the whole process is within a group. For a detailed discussion of this reimplementaion and its deficiencies, see pr/3160.

```

67 \def\add@accent#1#2{\hmode@bgroup

```

Turn off the group in `\UseTextSymbol` in case this is used inside the argument of `\add@accent`.

```

68   \let\hmode@start@before@group\@firstofone
69   \setbox\@tempboxa\hbox{#2%

```

When presetting the argument in a box we record its `\spacefactor` for later use after the accent got typeset. This way something like `\‘A` gets the spacefactor of A (i.e., 999) rather than the default value of 1000.

```

70     \global\mathchardef\accent@spacefactor\spacefactor}%
71   \accent#1 #2\egroup\spacefactor\accent@spacefactor}

```



Default definition for `\accent@spacefactor` prevents a horrible death of the above macro inside an unprotected `\edef`.

```
72 \let\accent@spacefactor\relax
```

```
\hmode\bgroup
```

```
73 \def\hmode\bgroup{\leavevmode\bgroup}
```

`\DeclareTextCompositeCommand` Another amusing game to play with `\expandafter`, `\csname`, and `\string`. When you say `\DeclareTextCompositeCommand{\foo}{T1}{a}{bar}`, we look to see if the expansion of `\T1\foo` begins with `\@text@composite`, and if it doesn't, we redefine `\T1\foo` to be:

```
\@text@composite
\@text@composite@x
\@strip@args
```

```
#1 -> \@text@composite \T1\foo #1\@empty \@text@composite {...}
```

where ... is the previous definition of `\T1\foo`. Finally, we define `\T1\foo-a` to expand to `bar`.

```
74 \</2ekernel>
75 \<latexrelease>\IncludeInRelease{2017/04/15}{\DeclareTextCompositeCommand}
76 \<latexrelease> {test for undeclared accent}%
77 \<*2ekernel | latexrelease>
78 \def\DeclareTextCompositeCommand#1#2#3#4{%
79   \expandafter\let\expandafter\reserved@a\csname#2\string#1\endcsname
80   \ifx\reserved@a\relax
81     \DeclareTextCommand#1{#2}{%
82       \@latex@error{\string#1 undeclared in encoding #2}\@eha}%
83       \@latex@info{Composite with undeclared \string#1 in encoding #2}%
84       \expandafter\let\expandafter\reserved@a\csname#2\string#1\endcsname
85     \fi
86     \expandafter\expandafter\expandafter\ifx
87     \expandafter\@car\reserved@a\relax\relax\@nil \@text@composite \else
88       \edef\reserved@b##1{%
89         \def\expandafter\noexpand
90           \csname#2\string#1\endcsname###1{%
91             \noexpand\@text@composite
92               \expandafter\noexpand\csname#2\string#1\endcsname
93               ###1\noexpand\@empty\noexpand\@text@composite
94               {##1}}}%
95       \expandafter\reserved@b\expandafter{\reserved@a{##1}}%
96     \fi
97     \expandafter\def\csname\expandafter\string\csname
98       #2\endcsname\string#1-\string#3\@empty\endcsname{#4}%
99   }
100 \</2ekernel | latexrelease>
101 \<latexrelease>\EndIncludeInRelease
102 \<latexrelease>\IncludeInRelease{0000/00/00}{\DeclareTextCompositeCommand}
103 \<latexrelease> {test for undeclared accent}%
104 \<latexrelease>\def\DeclareTextCompositeCommand#1#2#3#4{%
105 \<latexrelease> \expandafter\let\expandafter\reserved@a
106 \<latexrelease> \csname#2\string#1\endcsname
107 \<latexrelease> \expandafter\expandafter\expandafter\ifx
```



```

108 <latexrelease> \expandafter\@car\reserved@a\relax\relax\@nil
109 <latexrelease> \@text@composite \else
110 <latexrelease> \edef\reserved@b##1{%
111 <latexrelease> \def\expandafter\noexpand
112 <latexrelease> \csname#2\string#1\endcsname###1{%
113 <latexrelease> \noexpand\@text@composite
114 <latexrelease> \expandafter\noexpand\csname#2\string#1\endcsname
115 <latexrelease> ###1\noexpand\@empty\noexpand\@text@composite
116 <latexrelease> {##1}}}%
117 <latexrelease> \expandafter\reserved@b\expandafter{\reserved@a{##1}}%
118 <latexrelease> \fi
119 <latexrelease> \expandafter\def\csname\expandafter\string\csname
120 <latexrelease> #2\endcsname\string#1-\string#3\@empty\endcsname{#4}}
121 <latexrelease>\EndIncludeInRelease
122 <*2ekernel>
123 \@onlypreamble\DeclareTextCompositeCommand

```

This all works because:

```
\@text@composite \T1\foo A\@empty \@text@composite {...}
```

expands to `\T1\foo-A` if `\T1\foo-A` has been defined, and `{...}` otherwise.

Note that `\@text@composite` grabs the first token of the argument and puts just that in the csname. This is so that `\'\textit{e}` will work—it checks whether `\T1\'-\textit` is defined (which presumably it isn't) and so expands to `\accent 1 \textit{e}`.

This trick won't always work, for example `\'\{itshape e}` will expand to (with spaces added for clarity):

```
\csname \string \T1\' - \string {\itshape e} \@empty \endcsname
```

which will die pretty horribly. Unfortunately there's not much can be done about this if we're going to use `\csname` lookups as a fast way of accessing composites.

This has an unfortunate 'misfeature' though, which is that in the T1 encoding, `\'aa` produces á. This is not the expected behaviour, and should perhaps be fixed if the fix doesn't affect performance too badly.

Finally, it's worth noting that the `\@empty` is used in `\@text@composite` so that accents will work even when the argument is empty. If you say `\'{}` then this looks up `\T1\'-\@empty`, which ought to be `\relax`, and so all is well. If we didn't include the `\@empty`, then `\'{}` would expand to:

```
\csname \string \T1\' - \string \endcsname
```

so the `\endcsname` would be `\string'ed` and the whole of the rest of the document would be put inside the `\csname`. This would not be good.

```

124 \def\@text@composite#1#2#3\@text@composite{%
125   \expandafter\@text@composite@x
126   \csname\string#1-\string#2\endcsname}

```



Originally the `\@text@composite@x` macro had two arguments and if `#1` was not `\relax` it was executed, otherwise `#2` was executed. All this happened within the `\ifx` code so that neither `#1` nor `#2` could have picked up any additional arguments from the input stream. This has now been changed using the typical `\@firstoftwo / \@secondoftwo` coding. This way the final expansion will happen without any `\else` or `\fi` intervening in the case that we need to get a further token from the input stream.

```

127 \def\@text@composite@x#1{%
128   \ifx#1\relax
129     \expandafter\@secondoftwo
130   \else
131     \expandafter\@firstoftwo
132   \fi
133   #1}

```

The command `\DeclareTextComposite` uses `\DeclareTextCompositeCommand` to declare a command which expands out to a single glyph.

```

134 \catcode\z@=11\relax

135 \def\DeclareTextComposite#1#2#3#4{%
136   \def\reserved@a{\DeclareTextCompositeCommand#1{#2}{#3}}%
137   \bgroup
138     \lccode\z@#4%
139     \lowercase{%
140   \egroup
141   \reserved@a ^^@}}

142 \catcode\z@=15\relax

143 \@onlypreamble\DeclareTextComposite

```

`\UseTextAccent` These fragile commands access glyphs from different encodings. They use grotty  
`\UseTextSymbol` low-level calls to the font selection scheme for speed, and in order to make sure  
`\@use@text@encoding` that `\UseTextSymbol` doesn't do anything which you're not allowed to do between  
an `\accent` and its glyph.

For a detailed discussion of this reimplementaion and its deficiencies, see [pr/3160](#).

```

144 \def\UseTextAccent#1#2#3{%
145   \hmode@start@before@group
146   {%
147     \let\hmode@start@before@group\@firstofone
148     \let\@curr@enc\cf@encoding
149     \@use@text@encoding{#1}%
150     #2{\@use@text@encoding\@curr@enc#3}%
151   }}

152 \def\UseTextSymbol#1#2{%
153   \hmode@start@before@group

```



```

154      {%
155      \def\@wrong@font@char{\MessageBreak
156        for \noexpand\symbol'\string#2'}%
157      \@use@text@encoding{#1}%
158      #2%
159      }%
160    }

161 \def\@use@text@encoding#1{%
162   \edef\f@encoding{#1}%
163   \xdef\font@name{%
164     \csname\curr@fontshape/\f@size\endcsname}%
165   \pickup@font
166   \font@name
167   \@@enc@update}

```

`\hmode@start@before@group` The `\hmode@start@before@group` starts hmode and should be immediately followed by an explicit `{...}`. Its purpose is to ensure that hmode is started before this group is opened. Inside `\add@accent` and `\UseTextAccent` it is redefined to remove this group so that it doesn't conflict with the `\accent` primitive.

For a detailed discussion see pr/3160.

```
168 \let\hmode@start@before@group\leavevmode
```

`\DeclareTextSymbolDefault` Some syntactic sugar. Again, these should probably be optimized for speed.

```

\DeclareTextAccentDefault 169 \def\DeclareTextSymbolDefault#1#2{%
170   \DeclareTextCommandDefault#1{\UseTextSymbol{#2}#1}}

171 \def\DeclareTextAccentDefault#1#2{%
172   \DeclareTextCommandDefault#1{\UseTextAccent{#2}#1}}

173 \@onlypreamble\DeclareTextSymbolDefault
174 \@onlypreamble\DeclareTextAccentDefault

```

`\UndeclareTextCommand` This command safely removes an encoding specific declaration for a given encoding. It is helpful if one intends to use the default definition always and therefore wants to get rid of a declaration for some specific encoding.

```

175 \def\UndeclareTextCommand#1#2{%
    If there is no declaration for the current encoding do nothing. (This makes a hash
    table entry but without eTeX we can't do anything about that).

176   \expandafter\ifx\csname#2\string#1\endcsname\relax
177   \else
    Else: throw away that declaration.

178     \global\expandafter\let\csname#2\string#1\endcsname
179     \@undefined

```

But this is unfortunately not enough, we have to take a look at the top-level definition of the encoding specific command which for a command `\foo` would look similar to `\T1-cmd \foo \T1\foo` (three tokens).

Of course, instead of T1 one could see a different encoding name; which one depends the encoding for which `\foo` was declared last.



Now assume we have just removed the declaration for `\foo` in T1 and the top-level of `\foo` expands to the above. Then we better change that pretty fast otherwise we do get an “undefined csname error” when we try to typeset `\foo` within T1 instead of getting the default definition for `\foo`. And what is the best way to change that top-level definition? Well, the only “encoding” we know for sure will still be around is the default encoding denoted by `?`.

Thus in case the last token of the top-level expansion is now undefined we change the declaration to look like `\?-cmd \foo \?\foo` which is done by the following (readable?) code:

```

180     \expandafter\expandafter\expandafter
181     \ifx\expandafter\@thirdofthree#1\@undefined
182     \expandafter\gdef\expandafter#1\expandafter
183         {\csname ?-cmd\expandafter\endcsname\expandafter
184          #1\csname?\string#1\endcsname}%
185     \fi
186 \fi
187 }
188 \@onlypreamble\UndeclareTextCommand

```

### 1.4.2 Hyphenation

<code>\patterns</code> <code>\@@patterns</code> <code>\hyphenation</code> <code>\@@hyphenation</code>	<p>We redefine <code>\patterns</code> and <code>\hyphenation</code> to allow the use of commands declared with <code>\DeclareText*</code> to be used inside them.</p> <pre> 189 %\let\@@patterns\patterns 190 %\let\@@hyphenation\hyphenation 191 %\def\patterns{% 192 %    \bgroup 193 %        \let\protect\@empty 194 %        \let\@typeset@protect\@empty 195 %        \let\@changed@x\@changed@x@mouth 196 %    \afterassignment\egroup 197 %    \@@patterns 198 %} 199 %\def\hyphenation{% 200 %    \bgroup 201 %        \let\protect\@empty 202 %        \let\@typeset@protect\@empty 203 %        \let\@changed@x\@changed@x@mouth 204 %    \afterassignment\egroup 205 %    \@@hyphenation 206 %} </pre>
--	--

### 1.4.3 Miscellanea

- \a The `\a` command is used to access the accent commands even when they have been redefined (for example by the `tabbing` environment). Its internal name is `\@tabacckludge`.



The `\string` within the `\csname` guards against something like `'` being active at the point of use.

```
207 \def\@tabacckludge#1{\expandafter\@changed@cmd
208                               \csname\string#1\endcsname\relax}
209 \let\@a=\@tabacckludge
```

#### 1.4.4 Default encodings

We define the default encodings for most commands to be either OT1, OML or OMS. These defaults are in the kernel and therefore fonts with these encodings must be available unless these defaults are redefined elsewhere. Recall that the standard kernel loads the encoding files for these encodings, and also that for the T1 encoding.

The naming conventions in the kernel are not what we would use if we were starting from scratch... Those defined by DEK (like `\ae` and `\ss`) or by the T<sub>E</sub>X Users Group Technical Working Group on multi-lingual typesetting (like `\th` and `\ng`) have short names. Those which were added to the kernel in 1993 and early 1994 are named after their Adobe glyph names (like `\guillemotleft` and `\quotedblbase`). Unfortunately, this naming scheme won't work for all glyphs, since some names (like `\space`) are already used, and some (like `\endash`) are very likely to be defined by users. So we're now using the naming scheme of `\text` followed by the Adobe name, (like `\textendash` and `\textsterling`). Except that some glyphs don't have Adobe names, so we're using the names used by fontinst for those (like `\textcompwordmark`). Sigh.

Some accents from OT1:

```
210 \DeclareTextAccentDefault{"}{OT1}
211 \DeclareTextAccentDefault{'}{OT1}
212 \DeclareTextAccentDefault{.}{OT1}
213 \DeclareTextAccentDefault{=}{OT1}
214 \DeclareTextAccentDefault{H}{OT1}
215 \DeclareTextAccentDefault{^}{OT1}
216 \DeclareTextAccentDefault{'}{OT1}
217 \DeclareTextAccentDefault{b}{OT1}
218 \DeclareTextAccentDefault{c}{OT1}
219 \DeclareTextAccentDefault{d}{OT1}
220 \DeclareTextAccentDefault{r}{OT1}
221 \DeclareTextAccentDefault{u}{OT1}
222 \DeclareTextAccentDefault{v}{OT1}
223 \DeclareTextAccentDefault{~}{OT1}
```

Some symbols from OT1:

```
224 %\DeclareTextSymbolDefault{AA}{OT1}
225 \DeclareTextSymbolDefault{AE}{OT1}
226 \DeclareTextSymbolDefault{L}{OT1}
227 \DeclareTextSymbolDefault{OE}{OT1}
228 \DeclareTextSymbolDefault{O}{OT1}
229 %\DeclareTextSymbolDefault{aa}{OT1}
230 \DeclareTextSymbolDefault{ae}{OT1}
```



```

231 \DeclareTextSymbolDefault{\i}{OT1}
232 \DeclareTextSymbolDefault{\j}{OT1}

233 \DeclareTextSymbolDefault{\ij}{OT1}
234 \DeclareTextSymbolDefault{\IJ}{OT1}

235 \DeclareTextSymbolDefault{\l}{OT1}
236 \DeclareTextSymbolDefault{\oe}{OT1}
237 \DeclareTextSymbolDefault{\o}{OT1}
238 \DeclareTextSymbolDefault{\ss}{OT1}
239 \DeclareTextSymbolDefault{\textdollar}{OT1}
240 \DeclareTextSymbolDefault{\textemdash}{OT1}
241 \DeclareTextSymbolDefault{\textendash}{OT1}
242 \DeclareTextSymbolDefault{\textexclamdown}{OT1}
243 %\DeclareTextSymbolDefault{\texthyphenchar}{OT1}
244 %\DeclareTextSymbolDefault{\texthyphen}{OT1}
245 \DeclareTextSymbolDefault{\textquestiondown}{OT1}
246 \DeclareTextSymbolDefault{\textquotedblleft}{OT1}
247 \DeclareTextSymbolDefault{\textquotedblright}{OT1}
248 \DeclareTextSymbolDefault{\textquoteleft}{OT1}
249 \DeclareTextSymbolDefault{\textquoteright}{OT1}
250 \DeclareTextSymbolDefault{\textsterling}{OT1}

```

Some symbols from OMS:

```

251 \DeclareTextSymbolDefault{\textasteriskcentered}{OMS}
252 \DeclareTextSymbolDefault{\textbackslash}{OMS}
253 \DeclareTextSymbolDefault{\textbar}{OMS}
254 \DeclareTextSymbolDefault{\textbardbl}{OMS}
255 \DeclareTextSymbolDefault{\textbraceleft}{OMS}
256 \DeclareTextSymbolDefault{\textbraceright}{OMS}
257 \DeclareTextSymbolDefault{\textbullet}{OMS}
258 \DeclareTextSymbolDefault{\textdaggerdbl}{OMS}
259 \DeclareTextSymbolDefault{\textdagger}{OMS}
260 \DeclareTextSymbolDefault{\textparagraph}{OMS}
261 \DeclareTextSymbolDefault{\textperiodcentered}{OMS}
262 \DeclareTextSymbolDefault{\textsection}{OMS}
263 \DeclareTextAccentDefault{\textcircled}{OMS}

```

Some symbols from OML:

```

264 \DeclareTextSymbolDefault{\textless}{OML}
265 \DeclareTextSymbolDefault{\textgreater}{OML}
266 \DeclareTextAccentDefault{\t}{OML}

```

Some defaults we can fake.

The interface for defining `\copyright` changed, it used to use `\expandafter` to add braces at the appropriate points.

```

267 \DeclareTextCommandDefault{\textcopyright}{\textcircled{c}}
268 % \expandafter\def\expandafter
269 % \copyright\expandafter{\expandafter{\copyright}}

270 \DeclareTextCommandDefault{\textasciicircum}{\~{}}
271 \DeclareTextCommandDefault{\textasciitilde}{\~{}}

```



```

272 \DeclareTextCommandDefault{\textcompwordmark}{\leavevmode\kern\z@}
273 \DeclareTextCommandDefault{\textunderscore}{%
274   \leavevmode \kern.06em\vbox{\hrule\@width.3em}}

275 \DeclareTextCommandDefault{\textvisiblespace}{%
276   \mbox{\kern.06em\vrule \@height.3ex}%
277   \vbox{\hrule \@width.3em}%
278   \hbox{\vrule \@height.3ex}}

Using \fontdimen3 in the next definition is some sort of a kludge (since it
is the interword stretch) but it makes the ellipsis come out right in mono-spaced
fonts too (since there it is zero).

279 \DeclareTextCommandDefault{\textellipsis}{%
280   .\kern\fontdimen3\font
281   .\kern\fontdimen3\font
282   .\kern\fontdimen3\font}

283 %\DeclareTextCommandDefault{\textregistered}{\textcircled{\scshape r}}
284 \DeclareTextCommandDefault{\textregistered}{\textcircled{%
285   \check@mathfonts\fontsize\sf@size\z@\math@fontsfalse\selectfont R}}
286 \DeclareTextCommandDefault{\texttrademark}{\textsuperscript{TM}}
287 \DeclareTextCommandDefault{\SS}{SS}

288 \DeclareTextCommandDefault{\textordfeminine}{\textsuperscript{a}}
289 \DeclareTextCommandDefault{\textordmasculine}{\textsuperscript{o}}

```

#### 1.4.5 Math material

Some commands can be used in both text and math mode:

```

290 \DeclareRobustCommand{\$}{\ifmmode\mathdollar\else\textdollar\fi}
291 \DeclareRobustCommand{\{ }{\ifmmode\lbrace\else\textbraceleft\fi}
292 \DeclareRobustCommand{\} }{\ifmmode\rbrace\else\textbraceright\fi}
293 \DeclareRobustCommand{\P }{\ifmmode\mathparagraph\else\textparagraph\fi}
294 \DeclareRobustCommand{\S }{\ifmmode\mathsection\else\textsection\fi}
295 \DeclareRobustCommand{\dag }{\ifmmode\dagger\else\textdagger\fi}
296 \DeclareRobustCommand{\ddag }{\ifmmode\ddagger\else\textdaggerdbl\fi}

```

For historical reasons \copyright needs {} around the definition in maths.

```

297 \DeclareRobustCommand{\_ }{%
298   \ifmmode\nfss@text{\textunderscore}\else\textunderscore\fi}
299 \DeclareRobustCommand{\copyright }{%
300   \ifmmode{\nfss@text{\textcopyright}}\else\textcopyright\fi}
301 \DeclareRobustCommand{\pounds }{%
302   \ifmmode\mathsterling\else\textsterling\fi}

303 \DeclareRobustCommand{\dots }{%
304   \ifmmode\mathellipsis\else\textellipsis\fi}

305 \let\ldots\dots

```

Default definition of the commabelow accent.

```

306 (/2kernel)
307 <latexrelease>\IncludeInRelease{2015/10/01}{\textcommabelow}{comma accent}%

```



```

308 <*2ekernel | latexrelease>
309 \DeclareTextCommandDefault\textcommabelow[1]
310   {\hmode\bgroup\oalign{\null#1\crr\hidewidth\raise-.31ex
311     \hbox{\check@mathfonts\fontsize\ssf@size\z@
312       \math@fontsfalse\selectfont,}\hidewidth}\egroup}
313 <latexrelease>\EndIncludeInRelease
314 </2ekernel | latexrelease>
315 <latexrelease>\IncludeInRelease{0000/00/00}{\textcommabelow}{comma accent}%
316 <latexrelease>\let\textcommabelow\@undefined
317 <latexrelease>\expandafter
318 <latexrelease> \let\csname\string\T1\string\c-G\endcsname\@undefined
319 <latexrelease>\expandafter
320 <latexrelease> \let\csname\string\T1\string\c-K\endcsname\@undefined
321 <latexrelease>\expandafter
322 <latexrelease> \let\csname\string\T1\string\c-k\endcsname\@undefined
323 <latexrelease>\expandafter
324 <latexrelease> \let\csname\string\T1\string\c-L\endcsname\@undefined
325 <latexrelease>\expandafter
326 <latexrelease> \let\csname\string\T1\string\c-l\endcsname\@undefined
327 <latexrelease>\expandafter
328 <latexrelease> \let\csname\string\T1\string\c-N\endcsname\@undefined
329 <latexrelease>\expandafter
330 <latexrelease> \let\csname\string\T1\string\c-n\endcsname\@undefined
331 <latexrelease>\expandafter
332 <latexrelease> \let\csname\string\T1\string\c-R\endcsname\@undefined
333 <latexrelease>\expandafter
334 <latexrelease> \let\csname\string\T1\string\c-r\endcsname\@undefined
335 <latexrelease>\EndIncludeInRelease
    Default definition of the commaabove accent(E.G.).
336 <latexrelease>\IncludeInRelease{2016/02/01}{\textcommaabove}{comma above}%
337 <*2ekernel | latexrelease>
338 \DeclareTextCommandDefault\textcommaabove[1]{%
339   \hmode\bgroup
340   \oalign{%
341     \hidewidth
342     \raise.7ex\hbox{%
343       \check@mathfonts\fontsize\ssf@size\z@\math@fontsfalse\selectfont‘%
344     }%
345     \hidewidth\crr
346     \null#1\crr
347   }%
348   \egroup
349 }
350 <latexrelease>\EndIncludeInRelease
351 </2ekernel | latexrelease>
352 <latexrelease>\IncludeInRelease{0000/00/00}{\textcommaabove}{comma above}%
353 <latexrelease>\let\textcommaabove\@undefined
354 <latexrelease>\expandafter
355 <latexrelease> \let\csname\string\OT1\string\c-g\endcsname\@undefined

```



```

356 <latexrelease>\expandafter
357 <latexrelease> \let\csname\string\T1\string\c-g\endcsname\@undefined
358 <latexrelease>\EndIncludeInRelease

```

## 1.5 Definitions for the OT1 encoding

The definitions for the ‘T<sub>E</sub>X text’ (OT1) encoding.

Declare the encoding.

```

359 <*OT1>
360 \DeclareFontEncoding{OT1}{-}{-}

```

Declare the accents.

```

361 \DeclareTextAccent{"}{OT1}{127}
362 \DeclareTextAccent{'}{OT1}{19}
363 \DeclareTextAccent{.}{OT1}{95}
364 \DeclareTextAccent{=}{OT1}{22}
365 \DeclareTextAccent{^}{OT1}{94}
366 \DeclareTextAccent{'}{OT1}{18}
367 \DeclareTextAccent{~}{OT1}{126}
368 \DeclareTextAccent{H}{OT1}{125}
369 \DeclareTextAccent{u}{OT1}{21}
370 \DeclareTextAccent{v}{OT1}{20}
371 \DeclareTextAccent{r}{OT1}{23}

```

Some accents have to be built by hand: Note that \ooalign and \o@lign must be inside a group. In these definitions we no longer use the helper function \sh@ft from plain.tex since that now has two incompatible definitions.

```

372 \DeclareTextCommand{\b}{OT1}[1]
373   {\hmode\bgroup\o@lign{\relax#1\crrc\hidewidth\ltx@sh@ft{-3ex}%
374     \vbox to.2ex{\hbox{\char22}\vss}\hidewidth}\egroup}
375 \DeclareTextCommand{\c}{OT1}[1]
376   {\leavevmode\setbox\z@{\hbox{#1}\ifdim\ht\z@=1ex\accent24 #1%
377     \else\ooalign{\unhbox\z@\crrc\hidewidth\char24\hidewidth}}\fi}
378 \DeclareTextCommand{\d}{OT1}[1]
379   {\hmode\bgroup
380     \o@lign{\relax#1\crrc\hidewidth\ltx@sh@ft{-1ex}.\hidewidth}\egroup}

```

Declare the text symbols.

```

381 \DeclareTextSymbol{\AE}{OT1}{29}
382 \DeclareTextSymbol{\OE}{OT1}{30}
383 \DeclareTextSymbol{\O}{OT1}{31}
384 \DeclareTextSymbol{\ae}{OT1}{26}
385 \DeclareTextSymbol{\i}{OT1}{16}
386 \DeclareTextSymbol{\j}{OT1}{17}
387 \DeclareTextSymbol{\oe}{OT1}{27}
388 \DeclareTextSymbol{\o}{OT1}{28}
389 \DeclareTextSymbol{\ss}{OT1}{25}
390 \DeclareTextSymbol{\textendash}{OT1}{124}
391 \DeclareTextSymbol{\textendash}{OT1}{123}

```



Using the ligatures helps with OT1 fonts that have `\textexclamdown` and `\textquestiondown` in unusual positions.

```

392 %\DeclareTextSymbol{\textexclamdown}{OT1}{60}
393 %\DeclareTextSymbol{\textquestiondown}{OT1}{62}
394 \DeclareTextCommand{\textexclamdown}{OT1}{!}
395 \DeclareTextCommand{\textquestiondown}{OT1}{?}
396 %\DeclareTextSymbol{\texthyphenchar}{OT1}{\'}
397 %\DeclareTextSymbol{\texthyphen}{OT1}{\'}
398 \DeclareTextSymbol{\textquotedblleft}{OT1}{92}
399 \DeclareTextSymbol{\textquotedblright}{OT1}{93}
400 \DeclareTextSymbol{\textquoteleft}{OT1}{94}
401 \DeclareTextSymbol{\textquoteright}{OT1}{95}

```

Some symbols which are faked from others:

```

402 % \DeclareTextCommand{\aa}{OT1}
403 %   {\accent23a}
404 \DeclareTextCommand{\L}{OT1}
405   {\leavevmode\setbox\z@\hbox{L}\hb@xt@\wd\z@{\hss\@xxxii L}}
406 \DeclareTextCommand{\l}{OT1}
407   {\hmode\bgroup\@xxxii l\egroup}
408 % \DeclareTextCommand{\AA}{OT1}
409 %   {\leavevmode\setbox\z@\hbox{A}\dimen@ht\z@\advance\dimen@-1ex%
410 %     \rlap{\raise.67\dimen@\hbox{\char23}}A}

```

In the OT1 encoding Å has a hand-crafted definition, so we have here the first recorded explicit use of `\DeclareTextCompositeCommand`.

```

411 \DeclareTextCompositeCommand{\r}{OT1}{A}
412   {\leavevmode\setbox\z@\hbox{!}\dimen@ht\z@\advance\dimen@-1ex%
413     \rlap{\raise.67\dimen@\hbox{\char23}}A}

```

The dutch language uses the letter ‘ij’. It is available in T1 encoded fonts, but not in the OT1 encoded fonts. Therefor we fake it for the OT1 encoding.

```

414 \DeclareTextCommand{\ij}{OT1}{%
415   \nobreak\hskip\z@skip i\kern-0.02em j\nobreak\hskip\z@skip}
416 \DeclareTextCommand{\IJ}{OT1}{%
417   \nobreak\hskip\z@skip I\kern-0.02em J\nobreak\hskip\z@skip}

```

In the OT1 encoding, £ and \$ share a slot.

```

418 \DeclareTextCommand{\textdollar}{OT1}{\hmode\bgroup
419   \ifdim \fontdimen\@ne\font >\z@
420     \slshape
421   \else
422     \upshape
423   \fi
424   \char'\$ \egroup}

425 \DeclareTextCommand{\textsterling}{OT1}{\hmode\bgroup
426   \ifdim \fontdimen\@ne\font >\z@
427     \itshape
428   \else
429     \fontshape{ui}\selectfont

```



```

430 \fi
431 \char'\$ \egroup}

```

Here we are adding some more composite commands to the OT1 encoding. This makes the use of certain accents with `i` compatible with their use with the T1 encoding; this enables them to become true L<sup>A</sup>T<sub>E</sub>X internal representations. However, it will make these accents work a little less fast since a check will always be made for the existence of a composite.

```

432 \DeclareTextComposite{\.}{OT1}{i}{\i}
433 \DeclareTextComposite{\.}{OT1}{i}{\i}
434 \DeclareTextCompositeCommand{\'}{OT1}{i}{\@tabacckludge'\i}
435 \DeclareTextCompositeCommand{\'}{OT1}{i}{\@tabacckludge'\i}
436 \DeclareTextCompositeCommand{\^}{OT1}{i}{\^\i}
437 \DeclareTextCompositeCommand{\^}{OT1}{i}{\^\i}

```

T1 encoding is given more extensive set of overloads for `\c` But here we just adjust `\c{g}`.

```

438 \ifx\textcommaabove\undefined\else
439 \DeclareTextCompositeCommand{\c}{OT1}{g}{\textcommaabove{g}}
440 \fi
441 \</OT1>

```

## 1.6 Definitions for the T1 encoding

The definitions for the ‘Extended T<sub>E</sub>X text’ (T1) encoding.

Declare the encoding.

```

442 \<*T1>
443 \DeclareFontEncoding{T1}{}{}

```

Declare the accents.

```

444 \DeclareTextAccent{\'}{T1}{0}
445 \DeclareTextAccent{\'}{T1}{1}
446 \DeclareTextAccent{\^}{T1}{2}
447 \DeclareTextAccent{\^}{T1}{3}
448 \DeclareTextAccent{\^}{T1}{4}
449 \DeclareTextAccent{\H}{T1}{5}
450 \DeclareTextAccent{\r}{T1}{6}
451 \DeclareTextAccent{\v}{T1}{7}
452 \DeclareTextAccent{\u}{T1}{8}
453 \DeclareTextAccent{\=}{T1}{9}
454 \DeclareTextAccent{\.}{T1}{10}

```

Some accents have to be built by hand. Note that `\ooalign` and `\oalign` must be inside a group. In these definitions we no longer use the helper function `\sh@ft` from `plain.tex` since that now has two incompatible definitions.

```

455 \DeclareTextCommand{\b}{T1}[1]
456   {\hmode\bgroup\oalign{\relax#1\crrc\hidewidth\ltx@sh@ft{-3ex}%
457     \vbox to.2ex{\hbox{\char9}\vss}\hidewidth}\egroup}
458 \DeclareTextCommand{\c}{T1}[1]

```



```

459 {\leavevmode\setbox\z@\hbox{#1}\ifdim\ht\z@=1ex\accent11 #1%
460 \else\oalign{\unhbox\z@\crrc
461 \hidewidth\char11\hidewidth}}\fi}
462 \DeclareTextCommand{\d}{T1}[1]
463 {\hmode\bgroup
464 \o@lign{\relax#1\crrc\hidewidth\ltx@sh@ft{-1ex}.\hidewidth}\egroup}
465 \DeclareTextCommand{\k}{T1}[1]
466 {\hmode\bgroup\oalign{\null#1\crrc\hidewidth\char12}\egroup}
467 \DeclareTextCommand{\textogonekcentered}{T1}[1]
468 {\hmode\bgroup\oalign{%
469 \null#1\crrc\hidewidth\char12\hidewidth}\egroup}

```

Some symbols are constructed.

Slot 24 contains a small circle intended for construction of these two glyphs.

```

470 \DeclareTextCommand{\textperthousand}{T1}
471 {\%\char 24 } % space or 'relax as delimiter?
472 \DeclareTextCommand{\textpertenthousand}{T1}
473 {\%\char 24\char 24 } % space or 'relax as delimiter?

```

Declare the text symbols.

```

474 %\DeclareTextSymbol{\AA}{T1}{197}
475 \DeclareTextSymbol{\AE}{T1}{198}
476 \DeclareTextSymbol{\DH}{T1}{208}
477 \DeclareTextSymbol{\DJ}{T1}{208}
478 \DeclareTextSymbol{\L}{T1}{138}
479 \DeclareTextSymbol{\NG}{T1}{141}
480 \DeclareTextSymbol{\OE}{T1}{215}
481 \DeclareTextSymbol{\O}{T1}{216}
482 \DeclareTextSymbol{\SS}{T1}{223}
483 \DeclareTextSymbol{\TH}{T1}{222}
484 %\DeclareTextSymbol{\aa}{T1}{229}
485 \DeclareTextSymbol{\ae}{T1}{230}
486 \DeclareTextSymbol{\dh}{T1}{240}
487 \DeclareTextSymbol{\dj}{T1}{158}
488 \DeclareTextSymbol{\guillemotleft}{T1}{19}
489 \DeclareTextSymbol{\guillemotright}{T1}{20}
490 \DeclareTextSymbol{\guilsinglleft}{T1}{14}
491 \DeclareTextSymbol{\guilsinglright}{T1}{15}
492 \DeclareTextSymbol{\i}{T1}{25}
493 \DeclareTextSymbol{\j}{T1}{26}
494 \DeclareTextSymbol{\ij}{T1}{188}
495 \DeclareTextSymbol{\IJ}{T1}{156}
496 \DeclareTextSymbol{\l}{T1}{170}
497 \DeclareTextSymbol{\ng}{T1}{173}
498 \DeclareTextSymbol{\oe}{T1}{247}
499 \DeclareTextSymbol{\o}{T1}{248}
500 \DeclareTextSymbol{\quotedblbase}{T1}{18}
501 \DeclareTextSymbol{\quotesinglbase}{T1}{13}
502 \DeclareTextSymbol{\ss}{T1}{255}
503 \DeclareTextSymbol{\textasciicircum}{T1}{'\^}

```



```

504 \DeclareTextSymbol{\textasciitilde}{T1}{'\~}
505 \DeclareTextSymbol{\textbackslash}{T1}{'\}
506 \DeclareTextSymbol{\textbar}{T1}{'|}
507 \DeclareTextSymbol{\textbraceleft}{T1}{'\{ }
508 \DeclareTextSymbol{\textbraceright}{T1}{'\} }
509 \DeclareTextSymbol{\textcompwordmark}{T1}{23}
510 \DeclareTextSymbol{\textdollar}{T1}{'\$}
511 \DeclareTextSymbol{\textemdash}{T1}{22}
512 \DeclareTextSymbol{\textendash}{T1}{21}
513 \DeclareTextSymbol{\textexclamdown}{T1}{189}
514 \DeclareTextSymbol{\textgreater}{T1}{'\>}
515 %\DeclareTextSymbol{\texthyphenchar}{T1}{127}
516 %\DeclareTextSymbol{\texthyphen}{T1}{'\-}
517 \DeclareTextSymbol{\textless}{T1}{'\<}
518 \DeclareTextSymbol{\textquestiondown}{T1}{190}
519 \DeclareTextSymbol{\textquotedblleft}{T1}{16}
520 \DeclareTextSymbol{\textquotedblright}{T1}{17}
521 \DeclareTextSymbol{\textquotedbl}{T1}{'" }
522 \DeclareTextSymbol{\textquoteleft}{T1}{'\{ }
523 \DeclareTextSymbol{\textquoteright}{T1}{'\} }
524 \DeclareTextSymbol{\textsection}{T1}{159}
525 \DeclareTextSymbol{\textsterling}{T1}{191}
526 \DeclareTextSymbol{\textunderscore}{T1}{95}
527 \DeclareTextSymbol{\textvisiblespace}{T1}{32}
528 \DeclareTextSymbol{\th}{T1}{254}

```

Declare the composites.

```

529 \DeclareTextComposite{\.}{T1}{i}{'\i}
530 \DeclareTextComposite{\.}{T1}{\i}{'\i}

"80 = 128

531 \DeclareTextComposite{\u}{T1}{A}{128}
532 \DeclareTextComposite{\k}{T1}{A}{129}
533 \DeclareTextComposite{\'}{T1}{C}{130}
534 \DeclareTextComposite{\v}{T1}{C}{131}
535 \DeclareTextComposite{\v}{T1}{D}{132}
536 \DeclareTextComposite{\v}{T1}{E}{133}
537 \DeclareTextComposite{\k}{T1}{E}{134}
538 \DeclareTextComposite{\u}{T1}{G}{135}

"88 = 136

539 \DeclareTextComposite{\'}{T1}{L}{136}
540 \DeclareTextComposite{\v}{T1}{L}{137}
541 \DeclareTextComposite{\'}{T1}{N}{139}
542 \DeclareTextComposite{\v}{T1}{N}{140}
543 \DeclareTextComposite{\H}{T1}{O}{142}
544 \DeclareTextComposite{\'}{T1}{R}{143}

"90 = 144

545 \DeclareTextComposite{\v}{T1}{R}{144}
546 \DeclareTextComposite{\'}{T1}{S}{145}

```



```

547 \DeclareTextComposite{\v}{T1}{S}{146}
548 \DeclareTextComposite{\c}{T1}{S}{147}
549 \DeclareTextComposite{\v}{T1}{T}{148}
550 \DeclareTextComposite{\c}{T1}{T}{149}
551 \DeclareTextComposite{\H}{T1}{U}{150}
552 \DeclareTextComposite{\r}{T1}{U}{151}
"98 = 152
553 \DeclareTextComposite{\"}{T1}{Y}{152}
554 \DeclareTextComposite{\'}{T1}{Z}{153}
555 \DeclareTextComposite{\v}{T1}{Z}{154}
556 \DeclareTextComposite{\.}{T1}{Z}{155}
557 \DeclareTextComposite{\.}{T1}{I}{157}
"A0 = 160
558 \DeclareTextComposite{\u}{T1}{a}{160}
559 \DeclareTextComposite{\k}{T1}{a}{161}
560 \DeclareTextComposite{\'}{T1}{c}{162}
561 \DeclareTextComposite{\v}{T1}{c}{163}
562 \DeclareTextComposite{\v}{T1}{d}{164}
563 \DeclareTextComposite{\v}{T1}{e}{165}
564 \DeclareTextComposite{\k}{T1}{e}{166}
565 \DeclareTextComposite{\u}{T1}{g}{167}
"A8 = 168
566 \DeclareTextComposite{\'}{T1}{l}{168}
567 \DeclareTextComposite{\v}{T1}{l}{169}
568 \DeclareTextComposite{\'}{T1}{n}{171}
569 \DeclareTextComposite{\v}{T1}{n}{172}
570 \DeclareTextComposite{\H}{T1}{o}{174}
571 \DeclareTextComposite{\'}{T1}{r}{175}
"B0 = 176
572 \DeclareTextComposite{\v}{T1}{r}{176}
573 \DeclareTextComposite{\'}{T1}{s}{177}
574 \DeclareTextComposite{\v}{T1}{s}{178}
575 \DeclareTextComposite{\c}{T1}{s}{179}
576 \DeclareTextComposite{\v}{T1}{t}{180}
577 \DeclareTextComposite{\c}{T1}{t}{181}
578 \DeclareTextComposite{\H}{T1}{u}{182}
579 \DeclareTextComposite{\r}{T1}{u}{183}
"B8 = 184
580 \DeclareTextComposite{\"}{T1}{y}{184}
581 \DeclareTextComposite{\'}{T1}{z}{185}
582 \DeclareTextComposite{\v}{T1}{z}{186}
583 \DeclareTextComposite{\.}{T1}{z}{187}
"C0 = 192
584 \DeclareTextComposite{\'}{T1}{A}{192}
585 \DeclareTextComposite{\'}{T1}{A}{193}
586 \DeclareTextComposite{\^}{T1}{A}{194}

```



```

587 \DeclareTextComposite{\~}{T1}{A}{195}
588 \DeclareTextComposite{\"}{T1}{A}{196}
589 \DeclareTextComposite{\r}{T1}{A}{197}
590 \DeclareTextComposite{\c}{T1}{C}{199}
    "C8 = 200
591 \DeclareTextComposite{\'}{T1}{E}{200}
592 \DeclareTextComposite{\'}{T1}{E}{201}
593 \DeclareTextComposite{\~}{T1}{E}{202}
594 \DeclareTextComposite{\"}{T1}{E}{203}
595 \DeclareTextComposite{\'}{T1}{I}{204}
596 \DeclareTextComposite{\'}{T1}{I}{205}
597 \DeclareTextComposite{\~}{T1}{I}{206}
598 \DeclareTextComposite{\"}{T1}{I}{207}
    "D0 = 208
599 \DeclareTextComposite{\~}{T1}{N}{209}
600 \DeclareTextComposite{\'}{T1}{O}{210}
601 \DeclareTextComposite{\'}{T1}{O}{211}
602 \DeclareTextComposite{\~}{T1}{O}{212}
603 \DeclareTextComposite{\~}{T1}{O}{213}
604 \DeclareTextComposite{\"}{T1}{O}{214}
    "D8 = 216
605 \DeclareTextComposite{\'}{T1}{U}{217}
606 \DeclareTextComposite{\'}{T1}{U}{218}
607 \DeclareTextComposite{\~}{T1}{U}{219}
608 \DeclareTextComposite{\"}{T1}{U}{220}
609 \DeclareTextComposite{\'}{T1}{Y}{221}
    "E0 = 224
610 \DeclareTextComposite{\'}{T1}{a}{224}
611 \DeclareTextComposite{\'}{T1}{a}{225}
612 \DeclareTextComposite{\~}{T1}{a}{226}
613 \DeclareTextComposite{\~}{T1}{a}{227}
614 \DeclareTextComposite{\"}{T1}{a}{228}
615 \DeclareTextComposite{\r}{T1}{a}{229}
616 \DeclareTextComposite{\c}{T1}{c}{231}
    "E8 = 232
617 \DeclareTextComposite{\'}{T1}{e}{232}
618 \DeclareTextComposite{\'}{T1}{e}{233}
619 \DeclareTextComposite{\~}{T1}{e}{234}
620 \DeclareTextComposite{\"}{T1}{e}{235}
621 \DeclareTextComposite{\'}{T1}{i}{236}
622 \DeclareTextComposite{\'}{T1}{\i}{236}
623 \DeclareTextComposite{\'}{T1}{i}{237}
624 \DeclareTextComposite{\'}{T1}{\i}{237}
625 \DeclareTextComposite{\~}{T1}{i}{238}
626 \DeclareTextComposite{\~}{T1}{\i}{238}
627 \DeclareTextComposite{\"}{T1}{i}{239}
628 \DeclareTextComposite{\"}{T1}{\i}{239}

```



```

"FO = 240
629 \DeclareTextComposite{\~}{T1}{n}{241}
630 \DeclareTextComposite{\'}{T1}{o}{242}
631 \DeclareTextComposite{\'}{T1}{o}{243}
632 \DeclareTextComposite{\~}{T1}{o}{244}
633 \DeclareTextComposite{\~}{T1}{o}{245}
634 \DeclareTextComposite{\"}{T1}{o}{246}
"FO = 248
635 \DeclareTextComposite{\'}{T1}{u}{249}
636 \DeclareTextComposite{\'}{T1}{u}{250}
637 \DeclareTextComposite{\~}{T1}{u}{251}
638 \DeclareTextComposite{\"}{T1}{u}{252}
639 \DeclareTextComposite{\'}{T1}{y}{253}

640 \DeclareTextCompositeCommand{\k}{T1}{o}{\textogonekcentered{o}}
641 \DeclareTextCompositeCommand{\k}{T1}{0}{\textogonekcentered{0}}

642 \ifx\textcommaabove\undefined\else
643 \DeclareTextCompositeCommand{\c}{T1}{g}{\textcommaabove{g}}
644 \fi
645 \ifx\textcommabelow\undefined\else
646 \DeclareTextCompositeCommand{\c}{T1}{G}{\textcommabelow{G}}
647 \DeclareTextCompositeCommand{\c}{T1}{K}{\textcommabelow{K}}
648 \DeclareTextCompositeCommand{\c}{T1}{k}{\textcommabelow{k}}
649 \DeclareTextCompositeCommand{\c}{T1}{L}{\textcommabelow{L}}
650 \DeclareTextCompositeCommand{\c}{T1}{l}{\textcommabelow{l}}
651 \DeclareTextCompositeCommand{\c}{T1}{N}{\textcommabelow{N}}
652 \DeclareTextCompositeCommand{\c}{T1}{n}{\textcommabelow{n}}
653 \DeclareTextCompositeCommand{\c}{T1}{R}{\textcommabelow{R}}
654 \DeclareTextCompositeCommand{\c}{T1}{r}{\textcommabelow{r}}
655 \fi
656 \langle/T1\rangle

```

## 1.7 Definitions for the OMS encoding

The definitions for the ‘ $\TeX$  math symbol’ (OMS) encoding. Even though this is meant to be a math font, it includes some of the standard  $\LaTeX$  text symbols.

Declare the encoding.

```

657 \langle*OMS\rangle
658 \DeclareFontEncoding{OMS}{-}{-}

```

Declare the symbols. Note that slot 13 has in places been named `\Orb`: please root out and destroy this impolity wherever you find it!

```

659 \DeclareTextSymbol{\textasteriskcentered}{OMS}{3} % "03
660 \DeclareTextSymbol{\textbackslash}{OMS}{110} % "6E
661 \DeclareTextSymbol{\textbar}{OMS}{106} % "6A
662 \DeclareTextSymbol{\textbardbl}{OMS}{107} % "6B
663 \DeclareTextSymbol{\textbraceleft}{OMS}{102} % "66
664 \DeclareTextSymbol{\textbraceright}{OMS}{103} % "67

```



```

665 \DeclareTextSymbol{\textbullet}{OMS}{15} % "0F
666 \DeclareTextSymbol{\textdaggerdbl}{OMS}{122} % "7A
667 \DeclareTextSymbol{\textdagger}{OMS}{121} % "79
668 \DeclareTextSymbol{\textparagraph}{OMS}{123} % "7B
669 \DeclareTextSymbol{\textperiodcentered}{OMS}{1} % "01
670 \DeclareTextSymbol{\textsection}{OMS}{120} % "78
671 \DeclareTextSymbol{\textbigcircle}{OMS}{13} % "0D
672 \DeclareTextCommand{\textcircled}{OMS}[1]{\hmode@bgroup
673   \ooalign{%
674     \hfil \raise .07ex\hbox {\upshape#1}\hfil \crcr
675     \char 13 % "0D
676   }%
677 \egroup}
678 \end{OMS}

```

## 1.8 Definitions for the OML encoding

The definitions for the ‘ $\TeX$  math italic’ (OML) encoding. Even though this is meant to be a math font, it includes some of the standard  $\LaTeX$  text symbols.

Declare the encoding.

```

679 \begin{OML}
680 \DeclareFontEncoding{OML}{}{}
681
682 \DeclareTextSymbol{\textless}{OML}{'\<}
683 \DeclareTextSymbol{\textgreater}{OML}{'\>}
684 \DeclareTextAccent{\t}{OML}{127} % "7F
685 \end{OML}

```

## 1.9 Definitions for the OT4 encoding

These definitions are for the Polish extension to the ‘ $\TeX$  text’ (OT1) encoding. This encoding was created by B. Jackowski and M. Ryćko for use with the Polish version of Computer Modern and Computer Concrete. In positions 0–127 it is identical to OT1 but it contains some additional characters in the upper half. The  $\LaTeX$  support was developed by Mariusz Olko.

The PL fonts that use it are available as follows:

Metafont sources <ftp://ftp.gust.org.pl/TeX/language/polish/pl-mf.zip>;

Font files <ftp://ftp.gust.org.pl/TeX/language/polish/pl-tfm.zip>.

Declare the encoding.

```

685 \begin{OT4}
686 \DeclareFontEncoding{OT4}{}{}
687 \DeclareFontSubstitution{OT4}{cmr}{m}{n}
688
689 \DeclareTextAccent{"}{OT4}{127}
690 \DeclareTextAccent{'}{OT4}{19}
691 \DeclareTextAccent{\.}{OT4}{95}
692 \DeclareTextAccent{\=}{OT4}{22}

```



```

692 \DeclareTextAccent{\~}{OT4}{94}
693 \DeclareTextAccent{\'}{OT4}{18}
694 \DeclareTextAccent{\~}{OT4}{126}
695 \DeclareTextAccent{\H}{OT4}{125}
696 \DeclareTextAccent{\u}{OT4}{21}
697 \DeclareTextAccent{\v}{OT4}{20}
698 \DeclareTextAccent{\r}{OT4}{23}

```

The ogonek accent is available only under a e A & E. But we have to provide some definition for \k. Some other accents have to be built by hand as in OT1:

```

699 \DeclareTextCommand{\k}{OT4}[1]{%
700   \TextSymbolUnavailable{\k{#1}}#1}

```

In these definitions we no longer use the helper function \sh@ft from plain.tex since that now has two incompatible definitions.

```

701 \DeclareTextCommand{\b}{OT4}[1]
702   {\hmode\bgroup\o@lign{\relax#1\crrc\hidewidth\ltx@sh@ft{-3ex}%
703     \vbox to.2ex{\hbox{\char22}\vss}\hidewidth}\egroup}
704 \DeclareTextCommand{\c}{OT4}[1]
705   {\leavevmode\setbox\z@\hbox{#1}\ifdim\ht\z@=1ex\accent24 #1%
706     \else\o@lign{\unhbox\z@\crrc\hidewidth\char24\hidewidth}}\fi}
707 \DeclareTextCommand{\d}{OT4}[1]
708   {\hmode\bgroup
709     \o@lign{\relax#1\crrc\hidewidth\ltx@sh@ft{-1ex}.\hidewidth}\egroup}

```

Declare the text symbols.

```

710 \DeclareTextSymbol{\AE}{OT4}{29}
711 \DeclareTextSymbol{\OE}{OT4}{30}
712 \DeclareTextSymbol{\O}{OT4}{31}
713 \DeclareTextSymbol{\L}{OT4}{138}
714 \DeclareTextSymbol{\ae}{OT4}{26}
715 \DeclareTextSymbol{\guillemotleft}{OT4}{174}
716 \DeclareTextSymbol{\guillemotright}{OT4}{175}
717 \DeclareTextSymbol{\i}{OT4}{16}
718 \DeclareTextSymbol{\j}{OT4}{17}
719 \DeclareTextSymbol{\l}{OT4}{170}
720 \DeclareTextSymbol{\o}{OT4}{28}
721 \DeclareTextSymbol{\oe}{OT4}{27}
722 \DeclareTextSymbol{\quotedblbase}{OT4}{255}
723 \DeclareTextSymbol{\ss}{OT4}{25}
724 \DeclareTextSymbol{\textemdash}{OT4}{124}
725 \DeclareTextSymbol{\textendash}{OT4}{123}
726 \DeclareTextSymbol{\textexclamdown}{OT4}{60}
727 %\DeclareTextSymbol{\textthyphenchar}{OT4}{'\-}
728 %\DeclareTextSymbol{\textthyphen}{OT4}{'\-}
729 \DeclareTextSymbol{\textquestiondown}{OT4}{62}
730 \DeclareTextSymbol{\textquotedblleft}{OT4}{92}
731 \DeclareTextSymbol{\textquotedblright}{OT4}{'\"}
732 \DeclareTextSymbol{\textquoteleft}{OT4}{'\'}
733 \DeclareTextSymbol{\textquoteright}{OT4}{'\'}

```



Definition for Å as in OT1:

```
734 \DeclareTextCompositeCommand{\r}{OT4}{A}
735   {\leavevmode\setbox\z@\hbox{!}\dimen@ht\z@\advance\dimen@-1ex%
736    \rlap{\raise.67\dimen@\hbox{\char23}}A}
```

In the OT4 encoding, £ and \$ share a slot.

```
737 \DeclareTextCommand{\textdollar}{OT4}{\hmode@bgroup
738   \ifdim \fontdimen\@ne\font >\z@
739     \slshape
740   \else
741     \upshape
742   \fi
743   \char'\$\egroup}
744 \DeclareTextCommand{\textsterling}{OT4}{\hmode@bgroup
745   \ifdim \fontdimen\@ne\font >\z@
746     \itshape
747   \else
748     \fontshape{ui}\selectfont
749   \fi
750   \char'\$\egroup}
```

Declare the composites.

```
751 \DeclareTextComposite{\k}{OT4}{A}{129}
752 \DeclareTextComposite{\'}{OT4}{C}{130}
753 \DeclareTextComposite{\k}{OT4}{E}{134}
754 \DeclareTextComposite{\'}{OT4}{N}{139}
755 \DeclareTextComposite{\'}{OT4}{S}{145}
756 \DeclareTextComposite{\'}{OT4}{Z}{153}
757 \DeclareTextComposite{\.}{OT4}{Z}{155}
758 \DeclareTextComposite{\k}{OT4}{a}{161}
759 \DeclareTextComposite{\'}{OT4}{c}{162}
760 \DeclareTextComposite{\k}{OT4}{e}{166}
761 \DeclareTextComposite{\'}{OT4}{n}{171}
762 \DeclareTextComposite{\'}{OT4}{s}{177}
763 \DeclareTextComposite{\'}{OT4}{z}{185}
764 \DeclareTextComposite{\.}{OT4}{z}{187}
765 \DeclareTextComposite{\'}{OT4}{O}{211}
766 \DeclareTextComposite{\'}{OT4}{o}{243}
767 </OT4>
```

## 1.10 Definitions for the TS1 encoding

```
768 <*TS1>
769 \DeclareFontEncoding{TS1}{}{}
770 \DeclareFontSubstitution{TS1}{cmr}{m}{n}
```

Some accents have to be built by hand. Note that \ooalign and \o@lign must be inside a group.

```
771 \DeclareTextCommand{\capitalcedilla}{TS1}[1]
772   {\hmode@bgroup
773    \ooalign{\null#1\cr\hidewidth\char11\hidewidth}\egroup}
```



```

774 \DeclareTextCommand{\capitalogonek}{TS1}[1]
775   {\hmode\bgroup
776     \ooalign{\null#1\crrc\hidewidth\char12\hidewidth}\egroup}

```

Accents for capital letters.

These commands can be used by the end user either directly or through definitions of the type

```
\DeclareTextCompositeCommand{\'}{T1}{X}{\capitalacute X}
```

None of the latter definitions are provided by default, since they are probably rarely used.

"00 = 0

```

777 \DeclareTextAccent{\capitalgrave}{TS1}{0}
778 \DeclareTextAccent{\capitalacute}{TS1}{1}
779 \DeclareTextAccent{\capitalcircumflex}{TS1}{2}
780 \DeclareTextAccent{\capitaltilde}{TS1}{3}
781 \DeclareTextAccent{\capitaldieresis}{TS1}{4}
782 \DeclareTextAccent{\capitalhungarumlaut}{TS1}{5}
783 \DeclareTextAccent{\capitalring}{TS1}{6}
784 \DeclareTextAccent{\capitalcaron}{TS1}{7}

```

"08 = 8

```

785 \DeclareTextAccent{\capitalbreve}{TS1}{8}
786 \DeclareTextAccent{\capitalmacron}{TS1}{9}
787 \DeclareTextAccent{\capitaldotaccent}{TS1}{10}

```

Tie accents.

The tie accent was borrowed from the `cmmt` font. The `tc` fonts now provide four tie accents, the first two are done in the classical way with assymetric glyphs hanging out of their boxes; the new ties are centered in their boxes like all other accents. They need a name: please tell us if you know what to call them.

" =

```

788 \DeclareTextAccent{\t}{TS1}{26}
789 \DeclareTextAccent{\capitaltie}{TS1}{27}
790 \DeclareTextAccent{\newtie}{TS1}{28}
791 \DeclareTextAccent{\capitalnewtie}{TS1}{29}

```

Compound word marks.

The text companion fonts contain two compound word marks of different heights, one has `cap_height`, the other `asc_height`.

```

792 \DeclareTextSymbol{\textcapitalcompwordmark}{TS1}{23}
793 \DeclareTextSymbol{\textascendercompwordmark}{TS1}{31}

```

The text companion symbols.

```

794 \DeclareTextSymbol{\textquotestraightbase}{TS1}{13}

```

"10 = 16

```

795 \DeclareTextSymbol{\textquotestraightdblbase}{TS1}{18}
796 \DeclareTextSymbol{\texttwelveudash}{TS1}{21}
797 \DeclareTextSymbol{\textthreequartersemdash}{TS1}{22}

```



"18 = 24

```

798 \DeclareTextSymbol{\textleftarrow}{TS1}{24}
799 \DeclareTextSymbol{\textrightarrow}{TS1}{25}
"20 = 32
800 \DeclareTextSymbol{\textblank}{TS1}{32}
801 \DeclareTextSymbol{\textdollar}{TS1}{36}
802 \DeclareTextSymbol{\textquotesingle}{TS1}{39}
"28 = 40
803 \DeclareTextSymbol{\textasteriskcentered}{TS1}{42}
Note that '054 is a comma and '056 is a full stop: these make numbers using
oldstyle digits easier to input.
804 \DeclareTextSymbol{\textdblhyphen}{TS1}{45}
805 \DeclareTextSymbol{\textfractionsolidus}{TS1}{47}
Oldstyle digits.
"30 = 48
806 \DeclareTextSymbol{\textzerooldstyle}{TS1}{48}
807 \DeclareTextSymbol{\textoneoldstyle}{TS1}{49}
808 \DeclareTextSymbol{\texttwooldstyle}{TS1}{50}
809 \DeclareTextSymbol{\textthreeoldstyle}{TS1}{51}
810 \DeclareTextSymbol{\textfouroldstyle}{TS1}{52}
811 \DeclareTextSymbol{\textfiveoldstyle}{TS1}{53}
812 \DeclareTextSymbol{\textsixoldstyle}{TS1}{54}
813 \DeclareTextSymbol{\textsevenoldstyle}{TS1}{55}
"38 = 56
814 \DeclareTextSymbol{\texteightoldstyle}{TS1}{56}
815 \DeclareTextSymbol{\textnineoldstyle}{TS1}{57}
More text companion symbols.
816 \DeclareTextSymbol{\textlangle}{TS1}{60}
817 \DeclareTextSymbol{\textminus}{TS1}{61}
818 \DeclareTextSymbol{\textrangle}{TS1}{62}
"48 = 72
819 \DeclareTextSymbol{\textmho}{TS1}{77}
The big circle is here to define the command \textcircled. Formerly it was
taken from the cmsy font.
820 \DeclareTextSymbol{\textbigcircle}{TS1}{79}
821 \DeclareTextCommand{\textcircled}{TS1}[1]{\hmode@bgroup
822   \oalign{%
823     \hfil \raise .07ex\hbox {\upshape#1}\hfil \crcr
824     \char 79   % '117 = "4F
825   }%
826 \egroup}
More text companion symbols.
"50 = 80
827 \DeclareTextSymbol{\textohm}{TS1}{87}

```



"58 = 88

```

828 \DeclareTextSymbol{\textlbrackdbl}{TS1}{91}
829 \DeclareTextSymbol{\textrbrackdbl}{TS1}{93}
830 \DeclareTextSymbol{\textuparrow}{TS1}{94}
831 \DeclareTextSymbol{\textdownarrow}{TS1}{95}

```

"60 = 96

```

832 \DeclareTextSymbol{\textasciigrave}{TS1}{96}
833 \DeclareTextSymbol{\textborn}{TS1}{98}
834 \DeclareTextSymbol{\textdivorced}{TS1}{99}
835 \DeclareTextSymbol{\textdied}{TS1}{100}

```

"68 = 104

```

836 \DeclareTextSymbol{\textleaf}{TS1}{108}
837 \DeclareTextSymbol{\textmarried}{TS1}{109}
838 \DeclareTextSymbol{\textmusicalnote}{TS1}{110}

```

"78 = 120

```

839 \DeclareTextSymbol{\texttildelow}{TS1}{126}

```

This glyph, `\textdblhyphenchar` is hanging, like the hyphenchar of the ec fonts.

```

840 \DeclareTextSymbol{\textdblhyphenchar}{TS1}{127}

```

"80 = 128

```

841 \DeclareTextSymbol{\textasciibreve}{TS1}{128}
842 \DeclareTextSymbol{\textasciicaron}{TS1}{129}

```

This next glyph is *not* the same as `\textquotedbl`.

```

843 \DeclareTextSymbol{\textacutedbl}{TS1}{130}
844 \DeclareTextSymbol{\textgravedbl}{TS1}{131}
845 \DeclareTextSymbol{\textdagger}{TS1}{132}
846 \DeclareTextSymbol{\textdaggerdbl}{TS1}{133}
847 \DeclareTextSymbol{\textbardbl}{TS1}{134}
848 \DeclareTextSymbol{\textperthousand}{TS1}{135}

```

"88 = 136

```

849 \DeclareTextSymbol{\textbullet}{TS1}{136}
850 \DeclareTextSymbol{\textcelsius}{TS1}{137}
851 \DeclareTextSymbol{\textdollaroldstyle}{TS1}{138}
852 \DeclareTextSymbol{\textcentoldstyle}{TS1}{139}
853 \DeclareTextSymbol{\textflorin}{TS1}{140}
854 \DeclareTextSymbol{\textcolonmonetary}{TS1}{141}
855 \DeclareTextSymbol{\textwon}{TS1}{142}
856 \DeclareTextSymbol{\textnaira}{TS1}{143}

```

"90 = 144

```

857 \DeclareTextSymbol{\textguarani}{TS1}{144}
858 \DeclareTextSymbol{\textpeso}{TS1}{145}
859 \DeclareTextSymbol{\textlira}{TS1}{146}
860 \DeclareTextSymbol{\textrecipe}{TS1}{147}
861 \DeclareTextSymbol{\textinterrobang}{TS1}{148}

```



```

862 \DeclareTextSymbol{\textinterrobangdown}{TS1}{149}
863 \DeclareTextSymbol{\textdong}{TS1}{150}
864 \DeclareTextSymbol{\texttrademark}{TS1}{151}

```

"98 = 152

```

865 \DeclareTextSymbol{\textpertenthousand}{TS1}{152}
866 \DeclareTextSymbol{\textpilcrow}{TS1}{153}
867 \DeclareTextSymbol{\textbaht}{TS1}{154}
868 \DeclareTextSymbol{\textnumero}{TS1}{155}

```

This next name may change. For the following sign we know only a german name, which is abzüglich. The meaning is something like "commercial minus". An ASCII ersatz is ./ (dot slash dot). The temporary English name is \textdiscount.

```

869 \DeclareTextSymbol{\textdiscount}{TS1}{156}
870 \DeclareTextSymbol{\textestimated}{TS1}{157}
871 \DeclareTextSymbol{\textopenbullet}{TS1}{158}
872 \DeclareTextSymbol{\textservicemark}{TS1}{159}

```

"A0 = 160

```

873 \DeclareTextSymbol{\textlquill}{TS1}{160}
874 \DeclareTextSymbol{\textrquill}{TS1}{161}
875 \DeclareTextSymbol{\textcent}{TS1}{162}
876 \DeclareTextSymbol{\textsterling}{TS1}{163}
877 \DeclareTextSymbol{\textcurrency}{TS1}{164}
878 \DeclareTextSymbol{\textyen}{TS1}{165}
879 \DeclareTextSymbol{\textbrokenbar}{TS1}{166}
880 \DeclareTextSymbol{\textsection}{TS1}{167}

```

"A8 = 168

```

881 \DeclareTextSymbol{\textasciidieresis}{TS1}{168}
882 \DeclareTextSymbol{\textcopyright}{TS1}{169}
883 \DeclareTextSymbol{\textordfeminine}{TS1}{170}
884 \DeclareTextSymbol{\textcopyleft}{TS1}{171}
885 \DeclareTextSymbol{\textlnot}{TS1}{172}

```

The meaning of the circled-P is "sound recording copyright".

```

886 \DeclareTextSymbol{\textcircledP}{TS1}{173}
887 \DeclareTextSymbol{\textregistered}{TS1}{174}
888 \DeclareTextSymbol{\textasciimacron}{TS1}{175}

```

"B0 = 176

```

889 \DeclareTextSymbol{\textdegree}{TS1}{176}
890 \DeclareTextSymbol{\textpm}{TS1}{177}
891 \DeclareTextSymbol{\texttwosuperior}{TS1}{178}
892 \DeclareTextSymbol{\textthreesuperior}{TS1}{179}
893 \DeclareTextSymbol{\textasciacute}{TS1}{180}
894 \DeclareTextSymbol{\textmu}{TS1}{181} % micro sign
895 \DeclareTextSymbol{\textparagraph}{TS1}{182}
896 \DeclareTextSymbol{\textperiodcentered}{TS1}{183}

```

"B8 = 184

```

897 \DeclareTextSymbol{\textreferencemark}{TS1}{184}

```



```

898 \DeclareTextSymbol{\textonesuperior}{TS1}{185}
899 \DeclareTextSymbol{\textordmasculine}{TS1}{186}
900 \DeclareTextSymbol{\textsurd}{TS1}{187}
901 \DeclareTextSymbol{\textonequarter}{TS1}{188}
902 \DeclareTextSymbol{\textonehalf}{TS1}{189}
903 \DeclareTextSymbol{\textthreequarters}{TS1}{190}
904 \DeclareTextSymbol{\texteuro}{TS1}{191}
    "E0 = 208
905 \DeclareTextSymbol{\texttimes}{TS1}{214}
    "F0 = 240
906 \DeclareTextSymbol{\textdiv}{TS1}{246}
907 </TS1>

```

## 1.11 Definitions for the TU encoding

The TU encoding was originally introduced in the contributed package **fontspec** as a Unicode encoding for XeTeX and LuaTeX.

Normally for these engines, the input consists of Unicode characters encoded in UTF-8. There is therefore little need to use the traditional (ASCII) encoding-specific commands

However, sometimes (e.g. for backwards compatibility) it can be useful to access these Unicode characters via such ASCII-based markup. The commands provided here cover the characters in the T1 and TS1 encodings, but specified in Unicode position. Almost all the command names have been mechanically extracted from the **inputenc** UTF-8 support, which is essentially doing a reverse mapping from UTF-8 data to L<sup>A</sup>T<sub>E</sub>X LICR commands.

A few additional names for character which were supported in the original **fontspec** version of this file have also been added, even though they are not currently in the default **inputenc** UTF-8 declarations.

```

908 <*TU>

```

In the base interface the Unicode encoding is always known as TU. But we parameterise the encoding name to allow for modelling differences in Unicode support by different fonts.

```

909 \providecommand\UnicodeEncodingName{TU}

```

As the Unicode encoding, TU, is only currently available with XeTeX or LuaTeX, we detect these engines first, and make adjustments for the differing font loading syntax. For other engines, we issue a warning then abort this file, switching back to T1 encoding.

```

910 \begingroup\expandafter\expandafter\expandafter\endgroup
911 \expandafter\ifx\csname XeTeXrevision\endcsname\relax
912   \begingroup\expandafter\expandafter\expandafter\endgroup
913   \expandafter\ifx\csname directlua\endcsname\relax

```

Not LuaTeX or XeTeX, abort with a warning.

```

914   \PackageWarningNoLine{fontenc}

```



```

915     {\UnicodeEncodingName\space
916      encoding is only available with XeTeX and LuaTeX.\MessageBreak
917      Defaulting to T1 encoding}
918     \def\encodingdefault{T1}
919     \expandafter\expandafter\expandafter\endinput
920 \else
    LuaTeX.
921     \def\UnicodeFontTeXLigatures{+tlig;}
922     \def\reserved@a#1{%
923       \def\@remove@tlig##1{\@remove@tlig@##1\@nil#1\@nil\relax}
924       \def\@remove@tlig@##1#1{\@remove@tlig@##1}
925       \edef\reserved@b{\detokenize{+tlig;}}
926       \expandafter\reserved@a\expandafter{\reserved@b}
927       \def\@remove@tlig@##1\@nil#2\relax{#1}

928       \def\remove@tlig#1{%
929         \begingroup
930         \font\remove@tlig
931         \expandafter\@remove@tlig\expandafter{\fontname\font}%
932         \remove@tlig
933         \char#1\relax
934         \endgroup
935       }

936     \fi
937 \else
    XeTeX
938     \def\UnicodeFontTeXLigatures{mapping=tex-text;}
939     \def\remove@tlig#1{\XeTeXglyph\numexpr\XeTeXcharglyph#1\relax}
940 \fi

941 \def\UnicodeFontFile#1#2{"[#1]:#2"}
942 \def\UnicodeFontName#1#2{"#1:#2"}

    Declare the encoding
943 \DeclareFontEncoding\UnicodeEncodingName{}{}

    Declare accent command to use a postpended combining character rather than
    the TeX \accent primitive
944 \def\add@unicode@accent#1#2{%
945   \if\relax\detokenize{#2}\relax^^a0\else#2\fi
946   \char#1\relax}

947 \def\DeclareUnicodeAccent#1#2#3{%
948   \DeclareTextCommand{#1}{#2}{\add@unicode@accent{#3}}%
949 }

    Wrapper around \DeclareTextCompositeCommand that uses the declared com-
    posite if it exists in the current font or falls back to the default definition for the
    TU accent if not.

```



```

950 {
951 \catcode\z@=11\relax
952 \gdef\DeclareUnicodeComposite#1#2#3{%
953   \def\reserved@a##1##2{%
954     \DeclareTextCompositeCommand#1\UnicodeEncodingName{#2}{%
955       \iffontchar\font#3 ##2%
956       \else ##1\fi}}%
957   \expandafter\expandafter\expandafter\extract@default@composite
958   \csname\UnicodeEncodingName\string#1\endcsname{#2}\@nil
959   \bgroup
960     \lccode\z@#3 %
961     \lowercase{\egroup
962       \expandafter\reserved@a\expandafter{\reserved@b}{^^@}}}%
963 }

964 \def\extract@default@composite#1{%
965   \ifx\@text@composite#1%
966     \expandafter\extract@default@composite@a
967   \else
968     \expandafter\extract@default@composite@b\expandafter#1%
969   \fi}

970 \def\extract@default@composite@a#1\@text@composite#2\@nil{%
971   \def\reserved@b{#2}}
972 \def\extract@default@composite@b#1#2\@nil{%
973   \def\reserved@b{#1#2}}

974 \DeclareTextCommand\textquotesingle \UnicodeEncodingName{%
975   \remove@tlig{"0027}}
976 \DeclareTextCommand\textasciigrave \UnicodeEncodingName{%
977   \remove@tlig{"0060}}
978 \DeclareTextCommand\textquotedbl \UnicodeEncodingName{%
979   \remove@tlig{"0022}}

980 \DeclareTextSymbol{\textdollar} \UnicodeEncodingName{"0024}
981 \DeclareTextSymbol{\textless} \UnicodeEncodingName{"003C}
982 \DeclareTextSymbol{\textgreater} \UnicodeEncodingName{"003E}
983 \DeclareTextSymbol{\textbackslash} \UnicodeEncodingName{"005C}
984 \DeclareTextSymbol{\textasciicircum} \UnicodeEncodingName{"005E}
985 \DeclareTextSymbol{\textunderscore} \UnicodeEncodingName{"005F}
986 \DeclareTextSymbol{\textbraceleft} \UnicodeEncodingName{"007B}
987 \DeclareTextSymbol{\textbar} \UnicodeEncodingName{"007C}
988 \DeclareTextSymbol{\textbraceright} \UnicodeEncodingName{"007D}
989 \DeclareTextSymbol{\textasciitilde} \UnicodeEncodingName{"007E}
990 \DeclareTextSymbol{\textexclamdown} \UnicodeEncodingName{"00A1}
991 \DeclareTextSymbol{\textcent} \UnicodeEncodingName{"00A2}
992 \DeclareTextSymbol{\textsterling} \UnicodeEncodingName{"00A3}
993 \DeclareTextSymbol{\textcurrency} \UnicodeEncodingName{"00A4}
994 \DeclareTextSymbol{\textyen} \UnicodeEncodingName{"00A5}
995 \DeclareTextSymbol{\textbrokenbar} \UnicodeEncodingName{"00A6}
996 \DeclareTextSymbol{\textsection} \UnicodeEncodingName{"00A7}
997 \DeclareTextSymbol{\textasciidieresis} \UnicodeEncodingName{"00A8}

```



998 \DeclareTextSymbol{\textcopyright}	\UnicodeEncodingName{"00A9}
999 \DeclareTextSymbol{\textordfeminine}	\UnicodeEncodingName{"00AA}
1000 \DeclareTextSymbol{\guillemotleft}	\UnicodeEncodingName{"00AB}
1001 \DeclareTextSymbol{\textlnot}	\UnicodeEncodingName{"00AC}
1002 \DeclareTextSymbol{\textregistered}	\UnicodeEncodingName{"00AE}
1003 \DeclareTextSymbol{\textasciimacron}	\UnicodeEncodingName{"00AF}
1004 \DeclareTextSymbol{\textdegree}	\UnicodeEncodingName{"00B0}
1005 \DeclareTextSymbol{\textpm}	\UnicodeEncodingName{"00B1}
1006 \DeclareTextSymbol{\texttwosuperior}	\UnicodeEncodingName{"00B2}
1007 \DeclareTextSymbol{\textthreesuperior}	\UnicodeEncodingName{"00B3}
1008 \DeclareTextSymbol{\textasciicute}	\UnicodeEncodingName{"00B4}
1009 \DeclareTextSymbol{\textmu}	\UnicodeEncodingName{"00B5}
1010 \DeclareTextSymbol{\textparagraph}	\UnicodeEncodingName{"00B6}
1011 \DeclareTextSymbol{\textperiodcentered}	\UnicodeEncodingName{"00B7}
1012 \DeclareTextSymbol{\textonesuperior}	\UnicodeEncodingName{"00B9}
1013 \DeclareTextSymbol{\textordmasculine}	\UnicodeEncodingName{"00BA}
1014 \DeclareTextSymbol{\guillemotright}	\UnicodeEncodingName{"00BB}
1015 \DeclareTextSymbol{\textonequarter}	\UnicodeEncodingName{"00BC}
1016 \DeclareTextSymbol{\textonehalf}	\UnicodeEncodingName{"00BD}
1017 \DeclareTextSymbol{\textthreequarters}	\UnicodeEncodingName{"00BE}
1018 \DeclareTextSymbol{\textquestiondown}	\UnicodeEncodingName{"00BF}
1019 \DeclareTextSymbol{\AE}	\UnicodeEncodingName{"00C6}
1020 \DeclareTextSymbol{\DH}	\UnicodeEncodingName{"00D0}
1021 \DeclareTextSymbol{\texttimes}	\UnicodeEncodingName{"00D7}
1022 \DeclareTextSymbol{\O}	\UnicodeEncodingName{"00D8}
1023 \DeclareTextSymbol{\TH}	\UnicodeEncodingName{"00DE}
1024 \DeclareTextSymbol{\ss}	\UnicodeEncodingName{"00DF}
1025 \DeclareTextSymbol{\ae}	\UnicodeEncodingName{"00E6}
1026 \DeclareTextSymbol{\dh}	\UnicodeEncodingName{"00F0}
1027 \DeclareTextSymbol{\textdiv}	\UnicodeEncodingName{"00F7}
1028 \DeclareTextSymbol{\o}	\UnicodeEncodingName{"00F8}
1029 \DeclareTextSymbol{\th}	\UnicodeEncodingName{"00FE}
1030 \DeclareTextSymbol{\DJ}	\UnicodeEncodingName{"0110}
1031 \DeclareTextSymbol{\dj}	\UnicodeEncodingName{"0111}
1032 \DeclareTextSymbol{\i}	\UnicodeEncodingName{"0131}
1033 \DeclareTextSymbol{\IJ}	\UnicodeEncodingName{"0132}
1034 \DeclareTextSymbol{\ij}	\UnicodeEncodingName{"0133}
1035 \DeclareTextSymbol{\L}	\UnicodeEncodingName{"0141}
1036 \DeclareTextSymbol{\l}	\UnicodeEncodingName{"0142}
1037 \DeclareTextSymbol{\NG}	\UnicodeEncodingName{"014A}
1038 \DeclareTextSymbol{\ng}	\UnicodeEncodingName{"014B}
1039 \DeclareTextSymbol{\OE}	\UnicodeEncodingName{"0152}
1040 \DeclareTextSymbol{\oe}	\UnicodeEncodingName{"0153}
1041 \DeclareTextSymbol{\textflorin}	\UnicodeEncodingName{"0192}
1042 \DeclareTextSymbol{\j}	\UnicodeEncodingName{"0237}
1043 \DeclareTextSymbol{\textasciicaron}	\UnicodeEncodingName{"02C7}
1044 \DeclareTextSymbol{\textasciibreve}	\UnicodeEncodingName{"02D8}
1045 \DeclareTextSymbol{\textacutedbl}	\UnicodeEncodingName{"02DD}
1046 \DeclareTextSymbol{\textgravedbl}	\UnicodeEncodingName{"02F5}
1047 \DeclareTextSymbol{\texttildelow}	\UnicodeEncodingName{"02F7}



1048	\DeclareTextSymbol{\textbaht}	\UnicodeEncodingName{"0E3F}
1049	\DeclareTextSymbol{\SS}	\UnicodeEncodingName{"1E9E}
1050	\DeclareTextSymbol{\textcompwordmark}	\UnicodeEncodingName{"200C}
1051	\DeclareTextSymbol{\textendash}	\UnicodeEncodingName{"2013}
1052	\DeclareTextSymbol{\textemdash}	\UnicodeEncodingName{"2014}
1053	\DeclareTextSymbol{\textbardbl}	\UnicodeEncodingName{"2016}
1054	\DeclareTextSymbol{\textquoteleft}	\UnicodeEncodingName{"2018}
1055	\DeclareTextSymbol{\textquoteright}	\UnicodeEncodingName{"2019}
1056	\DeclareTextSymbol{\quotesinglbase}	\UnicodeEncodingName{"201A}
1057	\DeclareTextSymbol{\textquotedblleft}	\UnicodeEncodingName{"201C}
1058	\DeclareTextSymbol{\textquotedblright}	\UnicodeEncodingName{"201D}
1059	\DeclareTextSymbol{\quotedblbase}	\UnicodeEncodingName{"201E}
1060	\DeclareTextSymbol{\textdagger}	\UnicodeEncodingName{"2020}
1061	\DeclareTextSymbol{\textdaggerdbl}	\UnicodeEncodingName{"2021}
1062	\DeclareTextSymbol{\textbullet}	\UnicodeEncodingName{"2022}
1063	\DeclareTextSymbol{\textellipsis}	\UnicodeEncodingName{"2026}
1064	\DeclareTextSymbol{\textperthousand}	\UnicodeEncodingName{"2030}
1065	\DeclareTextSymbol{\textpertenthousand}	\UnicodeEncodingName{"2031}
1066	\DeclareTextSymbol{\guilsinglleft}	\UnicodeEncodingName{"2039}
1067	\DeclareTextSymbol{\guilsinglright}	\UnicodeEncodingName{"203A}
1068	\DeclareTextSymbol{\textreferencemark}	\UnicodeEncodingName{"203B}
1069	\DeclareTextSymbol{\textinterrobang}	\UnicodeEncodingName{"203D}
1070	\DeclareTextSymbol{\textfractionsolidus}	\UnicodeEncodingName{"2044}
1071	\DeclareTextSymbol{\textlquill}	\UnicodeEncodingName{"2045}
1072	\DeclareTextSymbol{\textrquill}	\UnicodeEncodingName{"2046}
1073	\DeclareTextSymbol{\textdiscount}	\UnicodeEncodingName{"2052}
1074	\DeclareTextSymbol{\textcolonmonetary}	\UnicodeEncodingName{"20A1}
1075	\DeclareTextSymbol{\textlira}	\UnicodeEncodingName{"20A4}
1076	\DeclareTextSymbol{\textnaira}	\UnicodeEncodingName{"20A6}
1077	\DeclareTextSymbol{\textwon}	\UnicodeEncodingName{"20A9}
1078	\DeclareTextSymbol{\textdong}	\UnicodeEncodingName{"20AB}
1079	\DeclareTextSymbol{\texteuro}	\UnicodeEncodingName{"20AC}
1080	\DeclareTextSymbol{\textpeso}	\UnicodeEncodingName{"20B1}
1081	\DeclareTextSymbol{\textcelsius}	\UnicodeEncodingName{"2103}
1082	\DeclareTextSymbol{\textnumero}	\UnicodeEncodingName{"2116}
1083	\DeclareTextSymbol{\textcircledP}	\UnicodeEncodingName{"2117}
1084	\DeclareTextSymbol{\textrecipe}	\UnicodeEncodingName{"211E}
1085	\DeclareTextSymbol{\textservicemark}	\UnicodeEncodingName{"2120}
1086	\DeclareTextSymbol{\texttrademark}	\UnicodeEncodingName{"2122}
1087	\DeclareTextSymbol{\textohm}	\UnicodeEncodingName{"2126}
1088	\DeclareTextSymbol{\textmho}	\UnicodeEncodingName{"2127}
1089	\DeclareTextSymbol{\textestimated}	\UnicodeEncodingName{"212E}
1090	\DeclareTextSymbol{\textleftarrow}	\UnicodeEncodingName{"2190}
1091	\DeclareTextSymbol{\textuparrow}	\UnicodeEncodingName{"2191}
1092	\DeclareTextSymbol{\textrightarrow}	\UnicodeEncodingName{"2192}
1093	\DeclareTextSymbol{\textdownarrow}	\UnicodeEncodingName{"2193}
1094	\DeclareTextSymbol{\textminus}	\UnicodeEncodingName{"2212}

Not all fonts have U+2217 but using U+002A requires some adjustment.

1095 \DeclareTextCommand{\textasteriskcentered}\UnicodeEncodingName{%



```

1096 \iffontchar\font"2217 \char"2217 \else
1097   \begingroup
1098     \fontsize
1099       {\the\dimexpr1.2\dimexpr\f@size pt\relax}%
1100       {\f@baselineskip}%
1101     \selectfont
1102     \raisebox{-0.6ex}{\dimexpr\height-0.6ex}[0pt]{*}%
1103   \endgroup
1104 \fi
1105 }

1106 \DeclareTextSymbol{\textsurd} \UnicodeEncodingName{"221A}
1107 \DeclareTextSymbol{\textlangle} \UnicodeEncodingName{"2329}
1108 \DeclareTextSymbol{\textrangle} \UnicodeEncodingName{"232A}
1109 \DeclareTextSymbol{\textblank} \UnicodeEncodingName{"2422}
1110 \DeclareTextSymbol{\textvisiblespace} \UnicodeEncodingName{"2423}
1111 \DeclareTextSymbol{\textopenbullet} \UnicodeEncodingName{"25E6}
1112 \DeclareTextSymbol{\textbigcircle} \UnicodeEncodingName{"25EF}
1113 \DeclareTextSymbol{\textmusicalnote} \UnicodeEncodingName{"266A}
1114 \DeclareTextSymbol{\textmarried} \UnicodeEncodingName{"26AD}
1115 \DeclareTextSymbol{\textdivorced} \UnicodeEncodingName{"26AE}
1116 \DeclareTextSymbol{\textinterrobangdown} \UnicodeEncodingName{"2E18}

Accents must be declared before the composites that use them.

1117 \DeclareUnicodeAccent{\` } \UnicodeEncodingName{"0300}
1118 \DeclareUnicodeAccent{\' } \UnicodeEncodingName{"0301}
1119 \DeclareUnicodeAccent{\^ } \UnicodeEncodingName{"0302}
1120 \DeclareUnicodeAccent{\~ } \UnicodeEncodingName{"0303}
1121 \DeclareUnicodeAccent{\= } \UnicodeEncodingName{"0304}
1122 \DeclareUnicodeAccent{\u } \UnicodeEncodingName{"0306}
1123 \DeclareUnicodeAccent{\. } \UnicodeEncodingName{"0307}
1124 \DeclareUnicodeAccent{\" } \UnicodeEncodingName{"0308}
1125 \DeclareUnicodeAccent{\r } \UnicodeEncodingName{"030A}
1126 \DeclareUnicodeAccent{\H } \UnicodeEncodingName{"030B}
1127 \DeclareUnicodeAccent{\v } \UnicodeEncodingName{"030C}
1128 \DeclareUnicodeAccent{\b } \UnicodeEncodingName{"0332}
1129 \DeclareUnicodeAccent{\d } \UnicodeEncodingName{"0323}
1130 \DeclareUnicodeAccent{\c } \UnicodeEncodingName{"0327}
1131 \DeclareUnicodeAccent{\k } \UnicodeEncodingName{"0328}
1132 \DeclareTextCommand\textcommabelow \UnicodeEncodingName[1]
1133 {\hmode\bgroup\oalign{\null#1\cr\hidewidth\raise-.31ex
1134 \hbox{\check@mathfonts\fontsize\ssf@size\z@
1135 \math@fontsfalse\selectfont,}\hidewidth}\egroup}

1136 \DeclareUnicodeComposite{\` } {"005E}
1137 \DeclareUnicodeComposite{\^ } {"007E}

1138 \DeclareUnicodeComposite{\` } {A}{"00C0}
1139 \DeclareUnicodeComposite{\' } {A}{"00C1}
1140 \DeclareUnicodeComposite{\^ } {A}{"00C2}
1141 \DeclareUnicodeComposite{\~ } {A}{"00C3}
1142 \DeclareUnicodeComposite{\" } {A}{"00C4}

```



1143 \DeclareUnicodeComposite{\r}	{A}{"00C5}
1144 \DeclareUnicodeComposite{\c}	{C}{"00C7}
1145 \DeclareUnicodeComposite{\'}	{E}{"00C8}
1146 \DeclareUnicodeComposite{\'}	{E}{"00C9}
1147 \DeclareUnicodeComposite{\^}	{E}{"00CA}
1148 \DeclareUnicodeComposite{\"}}	{E}{"00CB}
1149 \DeclareUnicodeComposite{\'}	{I}{"00CC}
1150 \DeclareUnicodeComposite{\'}	{I}{"00CD}
1151 \DeclareUnicodeComposite{\^}	{I}{"00CE}
1152 \DeclareUnicodeComposite{\"}}	{I}{"00CF}
1153 \DeclareUnicodeComposite{\~}	{N}{"00D1}
1154 \DeclareUnicodeComposite{\'}	{O}{"00D2}
1155 \DeclareUnicodeComposite{\'}	{O}{"00D3}
1156 \DeclareUnicodeComposite{\^}	{O}{"00D4}
1157 \DeclareUnicodeComposite{\~}	{O}{"00D5}
1158 \DeclareUnicodeComposite{\"}}	{O}{"00D6}
1159 \DeclareUnicodeComposite{\'}	{U}{"00D9}
1160 \DeclareUnicodeComposite{\'}	{U}{"00DA}
1161 \DeclareUnicodeComposite{\^}	{U}{"00DB}
1162 \DeclareUnicodeComposite{\"}}	{U}{"00DC}
1163 \DeclareUnicodeComposite{\'}	{Y}{"00DD}
1164 \DeclareUnicodeComposite{\'}	{a}{"00E0}
1165 \DeclareUnicodeComposite{\'}	{a}{"00E1}
1166 \DeclareUnicodeComposite{\^}	{a}{"00E2}
1167 \DeclareUnicodeComposite{\~}	{a}{"00E3}
1168 \DeclareUnicodeComposite{\"}}	{a}{"00E4}
1169 \DeclareUnicodeComposite{\r}	{a}{"00E5}
1170 \DeclareUnicodeComposite{\c}	{c}{"00E7}
1171 \DeclareUnicodeComposite{\'}	{e}{"00E8}
1172 \DeclareUnicodeComposite{\'}	{e}{"00E9}
1173 \DeclareUnicodeComposite{\^}	{e}{"00EA}
1174 \DeclareUnicodeComposite{\"}}	{e}{"00EB}
1175 \DeclareUnicodeComposite{\'}	\i {"00EC}
1176 \DeclareUnicodeComposite{\'}	{i}{"00EC}
1177 \DeclareUnicodeComposite{\'}	\i {"00ED}
1178 \DeclareUnicodeComposite{\'}	{i}{"00ED}
1179 \DeclareUnicodeComposite{\^}	\i {"00EE}
1180 \DeclareUnicodeComposite{\^}	{i}{"00EE}
1181 \DeclareUnicodeComposite{\"}}	\i {"00EF}
1182 \DeclareUnicodeComposite{\"}}	{i}{"00EF}
1183 \DeclareUnicodeComposite{\~}	{n}{"00F1}
1184 \DeclareUnicodeComposite{\'}	{o}{"00F2}
1185 \DeclareUnicodeComposite{\'}	{o}{"00F3}
1186 \DeclareUnicodeComposite{\^}	{o}{"00F4}
1187 \DeclareUnicodeComposite{\~}	{o}{"00F5}
1188 \DeclareUnicodeComposite{\"}}	{o}{"00F6}
1189 \DeclareUnicodeComposite{\'}	{u}{"00F9}
1190 \DeclareUnicodeComposite{\'}	{u}{"00FA}
1191 \DeclareUnicodeComposite{\^}	{u}{"00FB}
1192 \DeclareUnicodeComposite{\"}}	{u}{"00FC}



1193 \DeclareUnicodeComposite{\'}	{y}{"00FD}
1194 \DeclareUnicodeComposite{\"}	{y}{"00FF}
1195 \DeclareUnicodeComposite{\=}	{A}{"0100}
1196 \DeclareUnicodeComposite{\=}	{a}{"0101}
1197 \DeclareUnicodeComposite{\u}	{A}{"0102}
1198 \DeclareUnicodeComposite{\u}	{a}{"0103}
1199 \DeclareUnicodeComposite{\k}	{A}{"0104}
1200 \DeclareUnicodeComposite{\k}	{a}{"0105}
1201 \DeclareUnicodeComposite{\'}	{C}{"0106}
1202 \DeclareUnicodeComposite{\'}	{c}{"0107}
1203 \DeclareUnicodeComposite{\^}	{C}{"0108}
1204 \DeclareUnicodeComposite{\^}	{c}{"0109}
1205 \DeclareUnicodeComposite{\.}	{C}{"010A}
1206 \DeclareUnicodeComposite{\.}	{c}{"010B}
1207 \DeclareUnicodeComposite{\v}	{C}{"010C}
1208 \DeclareUnicodeComposite{\v}	{c}{"010D}
1209 \DeclareUnicodeComposite{\v}	{D}{"010E}
1210 \DeclareUnicodeComposite{\v}	{d}{"010F}
1211 \DeclareUnicodeComposite{\=}	{E}{"0112}
1212 \DeclareUnicodeComposite{\=}	{e}{"0113}
1213 \DeclareUnicodeComposite{\u}	{E}{"0114}
1214 \DeclareUnicodeComposite{\u}	{e}{"0115}
1215 \DeclareUnicodeComposite{\.}	{E}{"0116}
1216 \DeclareUnicodeComposite{\.}	{e}{"0117}
1217 \DeclareUnicodeComposite{\k}	{E}{"0118}
1218 \DeclareUnicodeComposite{\k}	{e}{"0119}
1219 \DeclareUnicodeComposite{\v}	{E}{"011A}
1220 \DeclareUnicodeComposite{\v}	{e}{"011B}
1221 \DeclareUnicodeComposite{\^}	{G}{"011C}
1222 \DeclareUnicodeComposite{\^}	{g}{"011D}
1223 \DeclareUnicodeComposite{\u}	{G}{"011E}
1224 \DeclareUnicodeComposite{\u}	{g}{"011F}
1225 \DeclareUnicodeComposite{\.}	{G}{"0120}
1226 \DeclareUnicodeComposite{\.}	{g}{"0121}
1227 \DeclareUnicodeComposite{\c}	{G}{"0122}
1228 \DeclareUnicodeComposite{\c}	{g}{"0123}
1229 \DeclareUnicodeComposite{\^}	{H}{"0124}
1230 \DeclareUnicodeComposite{\^}	{h}{"0125}
1231 \DeclareUnicodeComposite{\~}	{I}{"0128}
1232 \DeclareUnicodeComposite{\~}	\i {"0129}
1233 \DeclareUnicodeComposite{\~}	{i}{"0129}
1234 \DeclareUnicodeComposite{\=}	{I}{"012A}
1235 \DeclareUnicodeComposite{\=}	\i {"012B}
1236 \DeclareUnicodeComposite{\=}	{i}{"012B}
1237 \DeclareUnicodeComposite{\u}	{I}{"012C}
1238 \DeclareUnicodeComposite{\u}	\i {"012D}
1239 \DeclareUnicodeComposite{\u}	{i}{"012D}
1240 \DeclareUnicodeComposite{\k}	{I}{"012E}
1241 \DeclareUnicodeComposite{\k}	\i {"012F}
1242 \DeclareUnicodeComposite{\k}	{i}{"012F}



1243 \DeclareUnicodeComposite{.}	{I}{"0130}
1244 \DeclareUnicodeComposite{^}	{J}{"0134}
1245 \DeclareUnicodeComposite{\^}	\j {"0135}
1246 \DeclareUnicodeComposite{\^}	{j}{"0135}
1247 \DeclareUnicodeComposite{c}	{K}{"0136}
1248 \DeclareUnicodeComposite{c}	{k}{"0137}
1249 \DeclareUnicodeComposite{'}	{L}{"0139}
1250 \DeclareUnicodeComposite{'}	{l}{"013A}
1251 \DeclareUnicodeComposite{c}	{L}{"013B}
1252 \DeclareUnicodeComposite{c}	{l}{"013C}
1253 \DeclareUnicodeComposite{v}	{L}{"013D}
1254 \DeclareUnicodeComposite{v}	{l}{"013E}
1255 \DeclareUnicodeComposite{'}	{N}{"0143}
1256 \DeclareUnicodeComposite{'}	{n}{"0144}
1257 \DeclareUnicodeComposite{c}	{N}{"0145}
1258 \DeclareUnicodeComposite{c}	{n}{"0146}
1259 \DeclareUnicodeComposite{v}	{N}{"0147}
1260 \DeclareUnicodeComposite{v}	{n}{"0148}
1261 \DeclareUnicodeComposite{=}	{O}{"014C}
1262 \DeclareUnicodeComposite{=}	{o}{"014D}
1263 \DeclareUnicodeComposite{u}	{O}{"014E}
1264 \DeclareUnicodeComposite{u}	{o}{"014F}
1265 \DeclareUnicodeComposite{H}	{O}{"0150}
1266 \DeclareUnicodeComposite{H}	{o}{"0151}
1267 \DeclareUnicodeComposite{'}	{R}{"0154}
1268 \DeclareUnicodeComposite{'}	{r}{"0155}
1269 \DeclareUnicodeComposite{c}	{R}{"0156}
1270 \DeclareUnicodeComposite{c}	{r}{"0157}
1271 \DeclareUnicodeComposite{v}	{R}{"0158}
1272 \DeclareUnicodeComposite{v}	{r}{"0159}
1273 \DeclareUnicodeComposite{'}	{S}{"015A}
1274 \DeclareUnicodeComposite{'}	{s}{"015B}
1275 \DeclareUnicodeComposite{\^}	{S}{"015C}
1276 \DeclareUnicodeComposite{\^}	{s}{"015D}
1277 \DeclareUnicodeComposite{c}	{S}{"015E}
1278 \DeclareUnicodeComposite{c}	{s}{"015F}
1279 \DeclareUnicodeComposite{v}	{S}{"0160}
1280 \DeclareUnicodeComposite{v}	{s}{"0161}
1281 \DeclareUnicodeComposite{c}	{T}{"0162}
1282 \DeclareUnicodeComposite{c}	{t}{"0163}
1283 \DeclareUnicodeComposite{v}	{T}{"0164}
1284 \DeclareUnicodeComposite{v}	{t}{"0165}
1285 \DeclareUnicodeComposite{\~}	{U}{"0168}
1286 \DeclareUnicodeComposite{\~}	{u}{"0169}
1287 \DeclareUnicodeComposite{=}	{U}{"016A}
1288 \DeclareUnicodeComposite{=}	{u}{"016B}
1289 \DeclareUnicodeComposite{u}	{U}{"016C}
1290 \DeclareUnicodeComposite{u}	{u}{"016D}
1291 \DeclareUnicodeComposite{r}	{U}{"016E}
1292 \DeclareUnicodeComposite{r}	{u}{"016F}



1293 \DeclareUnicodeComposite{\H}	{U}{"0170}
1294 \DeclareUnicodeComposite{\H}	{u}{"0171}
1295 \DeclareUnicodeComposite{\k}	{U}{"0172}
1296 \DeclareUnicodeComposite{\k}	{u}{"0173}
1297 \DeclareUnicodeComposite{\^}	{W}{"0174}
1298 \DeclareUnicodeComposite{\^}	{w}{"0175}
1299 \DeclareUnicodeComposite{\^}	{Y}{"0176}
1300 \DeclareUnicodeComposite{\^}	{y}{"0177}
1301 \DeclareUnicodeComposite{\"}	{Y}{"0178}
1302 \DeclareUnicodeComposite{\'}	{Z}{"0179}
1303 \DeclareUnicodeComposite{\'}	{z}{"017A}
1304 \DeclareUnicodeComposite{\.}	{Z}{"017B}
1305 \DeclareUnicodeComposite{\.}	{z}{"017C}
1306 \DeclareUnicodeComposite{\v}	{Z}{"017D}
1307 \DeclareUnicodeComposite{\v}	{z}{"017E}
1308 \DeclareUnicodeComposite{\v}	{A}{"01CD}
1309 \DeclareUnicodeComposite{\v}	{a}{"01CE}
1310 \DeclareUnicodeComposite{\v}	{I}{"01CF}
1311 \DeclareUnicodeComposite{\v}	\i {"01D0}
1312 \DeclareUnicodeComposite{\v}	{i}{"01D0}
1313 \DeclareUnicodeComposite{\v}	{O}{"01D1}
1314 \DeclareUnicodeComposite{\v}	{o}{"01D2}
1315 \DeclareUnicodeComposite{\v}	{U}{"01D3}
1316 \DeclareUnicodeComposite{\v}	{u}{"01D4}
1317 \DeclareUnicodeComposite{\=}	\AE{"01E2}
1318 \DeclareUnicodeComposite{\=}	\ae{"01E3}
1319 \DeclareUnicodeComposite{\v}	{G}{"01E6}
1320 \DeclareUnicodeComposite{\v}	{g}{"01E7}
1321 \DeclareUnicodeComposite{\v}	{K}{"01E8}
1322 \DeclareUnicodeComposite{\v}	{k}{"01E9}
1323 \DeclareUnicodeComposite{\k}	{O}{"01EA}
1324 \DeclareUnicodeComposite{\k}	{o}{"01EB}
1325 \DeclareUnicodeComposite{\v}	\j {"01F0}
1326 \DeclareUnicodeComposite{\v}	{j}{"01F0}
1327 \DeclareUnicodeComposite{\'}	{G}{"01F4}
1328 \DeclareUnicodeComposite{\'}	{g}{"01F5}
1329 \DeclareUnicodeComposite{\textcommabelow}	{S}{"0218}
1330 \DeclareUnicodeComposite{\textcommabelow}	{s}{"0219}
1331 \DeclareUnicodeComposite{\textcommabelow}	{T}{"021A}
1332 \DeclareUnicodeComposite{\textcommabelow}	{t}{"021B}
1333 \DeclareUnicodeComposite{\=}	{Y}{"0232}
1334 \DeclareUnicodeComposite{\=}	{y}{"0232}
1335 \DeclareUnicodeComposite{\.}	{B}{"1E02}
1336 \DeclareUnicodeComposite{\.}	{b}{"1E03}
1337 \DeclareUnicodeComposite{\d}	{B}{"1E04}
1338 \DeclareUnicodeComposite{\d}	{b}{"1E05}
1339 \DeclareUnicodeComposite{\d}	{D}{"1E0C}
1340 \DeclareUnicodeComposite{\d}	{d}{"1E0D}
1341 \DeclareUnicodeComposite{\=}	{G}{"1E20}
1342 \DeclareUnicodeComposite{\=}	{g}{"1E21}



```

1343 \DeclareUnicodeComposite{\d}      {H}{"1E24}
1344 \DeclareUnicodeComposite{\d}      {h}{"1E25}
1345 \DeclareUnicodeComposite{\d}      {K}{"1E32}
1346 \DeclareUnicodeComposite{\d}      {k}{"1E33}
1347 \DeclareUnicodeComposite{\d}      {L}{"1E36}
1348 \DeclareUnicodeComposite{\d}      {l}{"1E37}
1349 \DeclareUnicodeComposite{\d}      {M}{"1E42}
1350 \DeclareUnicodeComposite{\d}      {m}{"1E43}
1351 \DeclareUnicodeComposite{\d}      {N}{"1E46}
1352 \DeclareUnicodeComposite{\d}      {n}{"1E47}
1353 \DeclareUnicodeComposite{\d}      {R}{"1E5A}
1354 \DeclareUnicodeComposite{\d}      {r}{"1E5B}
1355 \DeclareUnicodeComposite{\d}      {S}{"1E62}
1356 \DeclareUnicodeComposite{\d}      {s}{"1E63}
1357 \DeclareUnicodeComposite{\d}      {T}{"1E6C}
1358 \DeclareUnicodeComposite{\d}      {t}{"1E6D}
1359 \DeclareUnicodeComposite{\d}      {V}{"1E7E}
1360 \DeclareUnicodeComposite{\d}      {v}{"1E7F}
1361 \DeclareUnicodeComposite{\d}      {W}{"1E88}
1362 \DeclareUnicodeComposite{\d}      {w}{"1E89}
1363 \DeclareUnicodeComposite{\d}      {Z}{"1E92}
1364 \DeclareUnicodeComposite{\d}      {z}{"1E93}
1365 \DeclareUnicodeComposite{\d}      {A}{"1EA0}
1366 \DeclareUnicodeComposite{\d}      {a}{"1EA1}
1367 \DeclareUnicodeComposite{\d}      {E}{"1EB8}
1368 \DeclareUnicodeComposite{\d}      {e}{"1EB9}
1369 \DeclareUnicodeComposite{\d}      {I}{"1ECA}
1370 \DeclareUnicodeComposite{\d}      {i}{"1ECB}
1371 \DeclareUnicodeComposite{\d}      {O}{"1ECC}
1372 \DeclareUnicodeComposite{\d}      {o}{"1ECD}
1373 \DeclareUnicodeComposite{\d}      {U}{"1EE4}
1374 \DeclareUnicodeComposite{\d}      {u}{"1EE5}
1375 \DeclareUnicodeComposite{\d}      {Y}{"1EF4}
1376 \DeclareUnicodeComposite{\d}      {y}{"1EF5}
1377 </TU>

```

## 2 Package files

This file now also contains some packages that provide access to the more specialised encodings.

### 2.1 The fontenc package

This package allows authors to specify which encodings they will use. For each encoding F00, the package looks to see if the encoding F00 has already been declared. If it has not, the file `fooenc.def` is loaded. The default encoding is set to be F00.



In addition the package at the moment contains extra code to extend the `\@uclclist` (list of upper/lower case pairs) for encodings that involve cyrillic characters. THIS IS A TEMPORARY SOLUTION and will not stay this way forever (or so we hope) but right now we are missing a proper interface for this and didn't wanted to rush it.

1378 `\*package)`

Here we define a macro that extends the `\@uclclist` if needed and afterwards turns itself in a noop.

```

1379 \def\update@uclc@with@cyrillic{%
1380 \expandafter\def\expandafter\@uclclist\expandafter
1381 {\@uclclist
1382 \cyr\CYRA\cyrabhch\CYRABHCH\cyrabhchdsc\CYRABHCHDSC\cyrabhdze
1383 \CYRABHDZE\cyrabhha\CYRABHHA\cyr\CYRAE\cyrb\CYRB\cyrbyus
1384 \CYRBYUS\cyr\CYRC\cyrch\CYRCH\cyrchldsc\CYRCHLDSC\cyrchrdsc
1385 \CYRCHRDSC\cyrchvcrs\CYRCHVCRS\cyrd\CYRD\cyrdelta\CYRDELTA
1386 \cyrdje\CYRDJE\cyrdze\CYRDZE\cyrdzhe\CYRDZHE\cyre\CYRE\cyreps
1387 \CYREPS\cyrerev\CYREREV\cyrery\CYRERY\cyrf\CYRF\cyrfita
1388 \CYRFITA\cyrg\CYRG\cyrgdsc\CYRGDSC\cyrgdschcrs\CYRGDSCHCRS
1389 \cyrgchcrs\CYRGHCRS\cyrgkh\CYRGHK\cyrgup\CYRGUP\cyrh\CYRH
1390 \cyrhds\CYRHDSC\cyrhchcrs\CYRHCHCRS\cyrhkh\CYRHKH\cyrhdsn
1391 \CYRHRDSN\cyri\CYRI\cyrie\CYRIE\cyrii\CYRII\cyrishrt\CYRISHRT
1392 \cyrishrtdsc\CYRISHRTDSC\cyrizh\CYRIZH\cyrje\CYRJE\cyrk\CYRK
1393 \cyrkbeak\CYRKBEAK\cyrkds\CYRKDSC\cyrkchcrs\CYRKCHCRS\cyrkhh
1394 \CYRKHK\cyrkvcrs\CYRKVCRS\cyr\CYRL\cyrldsc\CYRLDSC\cyr\lkh
1395 \CYRLHK\cyr\lje\CYRLJE\cyr\CYRM\cyrmdsc\CYRMDSC\cyr\mhk\CYRMHK
1396 \cyr\CYRN\cyrndsc\CYRNDSC\cyr\CYRNG\cyrnhk\CYRNHK\cyrnje
1397 \CYRNJE\cyrnlhk\CYRNLHK\cyro\CYRO\cyrotld\CYROTLD\cyrp\CYRP
1398 \cyrphk\CYRPHK\cyrq\CYRQ\cyrr\CYRR\cyrrdsc\CYRRDSC\cyrrhk
1399 \CYRRHK\cyrrtick\CYRRTICK\cyrs\CYRS\cyrsacrs\CYRSACRS
1400 \cyrschwa\CYRSCHWA\cyrsdsc\CYRSDSC\cyrsemisftsn\CYRSEMISFTSN
1401 \cyrsftsn\CYRSFTSN\cyrrsh\CYRSH\cyrrshch\CYRSHCH\cyrrshha\CYRSHHA
1402 \cyrt\CYRT\cyrtsc\CYRTDSC\cyrtetse\CYRTETSE\cyrtshe\CYRTSHE
1403 \cyr\CYRU\cyrushrt\CYRUSHRT\cyrv\CYRV\cyrw\CYRW\cyr\CYRY
1404 \CYRYA\CYRYA\cyr\CYRYAT\CYRYAT\cyr\cyryhcrs\CYRYHCRS\cyryi\CYRYI\cyryo
1405 \CYRYO\cyr\CYRYU\CYRYU\cyrz\CYRZ\cyrzdsc\CYRZDSC\cyrrzh\CYRZH
1406 \cyrrzhdsc\CYRZHDSC}%
1407 \let\update@uclc@with@cyrillic\relax
1408 }

```

Here we process each option:

```

1409 \DeclareOption*{%
1410 \let\encodingdefault\CurrentOption
1411 \edef\reserved@f{%
1412 \lowercase{\def\noexpand\reserved@f{\CurrentOption enc.def}}}%
1413 \reserved@f
1414 \InputIfFileExists\reserved@f
1415 {}{\PackageError{fontenc}%
1416 {Encoding file '\reserved@f' not found.%
1417 \MessageBreak

```



```

1418         You might have misspelt the name of the encoding}%
1419     {Necessary code for this encoding was not
1420     loaded.\MessageBreak
1421     Thus calling the encoding later on will
1422     produce further error messages.}}%
1423 \let\reserved@f\relax

    In case the current encoding is one of a list of known cyrillic ones we extend
    the \@uclclist:
1424 \expandafter\in@\expandafter{\CurrentOption}%
1425                               {T2A,T2B,T2C,X2,LCY,OT2}%
1426 \ifin@

    But only if it hasn't already been extended. This might happen if there are
    several calls to fontenc loading one of the above encodings. If we don't do this check
    the \@uclclist gets unnecessarily big, slowing down the processing at runtime.
1427 \expandafter\in@\expandafter\cyr@\expandafter
1428                               {\@uclclist}%
1429 \ifin@
1430 \else
1431     \update@uclc@with@cyrillic
1432 \fi
1433 \fi
1434 }

1435 \ProcessOptions*
1436 \fontencoding\encodingdefault\selectfont

    To save some space we get rid of the macro extending the \@uclclist (might
    have happened already).
1437 \let\update@uclc@with@cyrillic\relax

    Finally we pretend that the fontenc package wasn't read in. This allows for
    using it several times, e.g., in a class file and in the preamble (at the cost of not
    getting any version info). That kind of hackery shows that using a general purpose
    package just for loading an encoding is not the right kind of interface for setting
    up encodings — it will get replaced at some point in the future.
1438 \global\expandafter\let\csname ver@fontenc.sty\endcsname\relax
1439 \global\expandafter\let\csname opt@fontenc.sty\endcsname\relax
1440 \global\let\@ifl@ter@@\@ifl@ter
1441 \def\@ifl@ter#1#2#3#4#5{\global\let\@ifl@ter\@ifl@ter@@}
1442 \end{package}

```

## 2.2 The textcomp package

This one is for the TS1 encoding which contains text symbols for use with the T1-encoded text fonts. It therefore first inputs the file `TS1enc.def` and then sets (or resets) the defaults for the symbols it contains. The result of this is that when one of these symbols is accessed and the current encoding does not provide it, the symbol will be supplied by a silent, local change to this encoding.



1443  $\langle *TS1sty \rangle$

Since many PostScript fonts only implement a subset of **TS1** many commands only produce black blobs of ink. To resolve the resulting problems a number of options have been introduced and some code has been developed to distinguish sub-encodings.

The sub-encodings have a numerical id and are defined as follows for **TS1**:

**#5** those **TS1** symbols that are also in the ISO-Adobe character set; without **textcurrency**, which is often misused for the Euro. Older Type1 fonts from the non-**T<sub>E</sub>X** world provide only this subset.

**#4** = **#5** + **\texteuro**. Most newer fonts provide this.

**#3** = **#4** + **\textomega**. Can also be described as  $TS1 \cap (ISO-Adobe \cup MacRoman)$ . (Except for the missing "currency".)

**#2** = **#3** + **\textestimated** + **\textcurrency**. Can also be described as  $TS1 \cap Adobe-Western-2$ . This may be relevant for OpenType fonts, which usually show the Adobe-Western-2 character set.

**#1** = **TS1** without **\textcircled** and **\t**. These two glyphs are often not implemented and if their kernel defaults are changed commands like **\copyright** unnecessarily fail.

**#0** = full **TS1**

And here a summary to go in the transcript file:

```
1444 \PackageInfo{textcomp}{Sub-encoding information:\MessageBreak
1445   \space\space 5 = only ISO-Adobe without
1446                               \string\textcurrency\MessageBreak
1447   \space\space 4 = 5 + \string\texteuro\MessageBreak
1448   \space\space 3 = 4 + \string\textohm\MessageBreak
1449   \space\space 2 = 3 + \noexpand\textestimated+
1450                               \string\textcurrency\MessageBreak
1451   \space\space 1 = TS1 - \noexpand\textcircled-
1452                               \string\t\MessageBreak
1453   \space\space 0 = TS1 (full)\MessageBreak
1454   Font families with sub-encoding setting implement\MessageBreak
1455   only a restricted character set as indicated.\MessageBreak
1456   Family '?' is the default used for unknown fonts.\MessageBreak
1457   See the documentation for details\@gobble}
```

**\DeclareEncodingSubset** An encoding subset to which a font family belongs is declared by the command **\DeclareEncodingSubset** that takes the major encoding as the first argument (e.g., **TS1**), the family name as the second argument (e.g., **cmr**), and the subset encoding id as a third, (e.g., 0 for **cmr**).

The default encoding subset to use when nothing is known about the current font family is named **?nn**.

```
1458 \def\DeclareEncodingSubset#1#2#3{%
```



```

1459 \@ifundefined{#1:#2}%
1460   {\PackageInfo{textcomp}{Setting #2 sub-encoding to #1/#3}}%
1461   {\PackageInfo{textcomp}{Changing #2 sub-encoding to #1/#3}}%
1462   \namedef{#1:#2}{#3}}
1463 \@onlypreamble\DeclareEncodingSubset

```

The options for the package are the following:

**safe** for unknown font families enables only symbols that are also in the ISO-Adobe character set; without “currency”, which is often misused for the Euro. Older Type1 fonts from the non-TeX world provide only this subset.

**euro** enables the “safe” symbols plus the `\texteuro` command. Most newer fonts provide this.

**full** enables all TS1 commands; useful only with fonts like EC or CM bright.

**almostfull** same as “full”, except that `\textcircled` and `\t` are *not* redefined from their defaults to avoid that commands like `\copyright` suddenly no longer work.

**force** ignore all subset encoding definitions stored in the package itself or in the configuration file and always use the default subset as specified by one of the other options (seldom useful, only dangerous).

`\iftc@forced` Switch used to implement the **force** option

```

1464 \newif\iftc@forced \tc@forcedfalse

```

This is implemented by defining the default subset:

```

1465 \DeclareOption{full}{\DeclareEncodingSubset{TS1}{?}{0}}
1466 \DeclareOption{almostfull}{\DeclareEncodingSubset{TS1}{?}{1}}
1467 \DeclareOption{euro}{\DeclareEncodingSubset{TS1}{?}{4}}
1468 \DeclareOption{safe}{\DeclareEncodingSubset{TS1}{?}{5}}

```

The default is “almostfull” which means that old documents will work except that `\textcircled` and `\t` will use the kernel defaults (with the advantage that this also works if the current font (as often the case) doesn’t implement these glyphs.

The “force” option simply sets the switch to true.

```

1469 \DeclareOption{force}{\tc@forcedtrue}

```

The suggestions to user is to use the “safe” option always unless that balks in which case they could switch to “almostfull” but then better check their output manually.

```

1470 \def\tc@errorwarn{\PackageError}
1471 \DeclareOption{warn}{\gdef\tc@errorwarn#1#2#3{\PackageWarning{#1}{#2}}}
1472 \ExecuteOptions{almostfull}
1473 \ProcessOptions\relax

```



`\CheckEncodingSubset` The command `\CheckEncodingSubset` will check if the current font family has the right encoding subset to typeset a certain command. It takes five arguments as follows: first argument is either `\UseTextSymbol`, `\UseTextAccent` depending on whether or not the symbol is a text symbol or a text accent.

The second argument is the encoding from which this symbol should be fetched.

The third argument is either a fake accessor command or an error message. the code in that argument (if ever executed) receives two arguments: `#2` and `#5` of `\CheckEncodingSubset`.

Argument four is the subset encoding id to test against: if this value is higher than the subset id of the current font family then we typeset the symbol, i.e., execute `#1{#2}#5` otherwise it runs `#3#5`, e.g., to produce an error message or fake the glyph somehow.

Argument five is the symbol or accent command that is being checked.

For usage examples see definitions below.

1474 `\iftrue\forced`

If the “force” option was given we always use the default for testing against.

```
1475 \def\CheckEncodingSubset#1#2#3#4#5{%
1476   \ifnum #4>%
1477     0\csname #2:?\endcsname
1478   \relax
1479   \expandafter\@firstoftwo
1480 \else
1481   \expandafter\@secondoftwo
1482 \fi
1483   {#1{#2}}{#3}%
1484   #5%
1485 }
```

In normal circumstances the test is a bit more complicated: first check if there exists a macro `\langle arg2\rangle:\langle current-family\rangle` and if so use that value to test against, otherwise use the default to test against.

```
1486 \else
1487 \def\CheckEncodingSubset#1#2#3#4#5{%
1488   \ifnum #4>%
1489     \expandafter\ifx\csname #2:\f@family\endcsname\relax
1490       0\csname #2:?\endcsname
1491     \else
1492       \csname #2:\f@family\endcsname
1493     \fi
1494   \relax
1495   \expandafter\@firstoftwo
1496 \else
1497   \expandafter\@secondoftwo
1498 \fi
1499   {#1{#2}}{#3}%
1500   #5%
1501 }
1502 \fi
```



tc@subst

```
1503 \def\tc@subst#1{%
1504   \tc@errorwarn{textcomp}% % should be latex error if general
1505   {Symbol \string#1 not provided by\MessageBreak
1506    font family \f@family\space
1507    in TS1 encoding.\MessageBreak Default family used instead}\@eha
1508   \bgroup\fontfamily\textcompsubstdefault\selectfont#1\egroup
1509 }
```

\textcompsubstdefault

```
1510 \def\textcompsubstdefault{cmr}
```

\tc@error \tc@error is going to be used in arg #3 of \CheckEncodingSubset when a symbol is not available in a certain font family. It gets pass the encoding it normally lives in (arg one) and the name of the symbol or accent that has a problem.

```
1511 % error commands take argument:
1512 % #1 symbol to be used
1513 \def\tc@error#1{%
1514   \PackageError{textcomp}% % should be latex error if general
1515   {Accent \string#1 not provided by\MessageBreak
1516    font family \f@family\space
1517    in TS1 encoding}\@eha
1518 }
```

\tc@fake@euro \tc@fake@euro is an example of a “fake” definition to use in arg #3 of \CheckEncodingSubset when a symbol is not available in a certain font family. Here we produce an Euro symbol by combining a “C” with a “=”.

```
1519 \def\tc@fake@euro#1{%
1520   \leavevmode
1521   \PackageInfo{textcomp}{Faking \noexpand#1for font family
1522                                \f@family\MessageBreak in TS1 encoding}%
1523   \valign{##\cr
1524     \vfil\hbox to 0.07em{\dimen@ \f@size\p@
1525                               \math@fontsfalse
1526                               \fontsize{.7\dimen@}\z@\selectfont=\hss}%
1527     \vfil\cr%
1528     \hbox{C}\crrc
1529   }%
1530 }
```

\tc@check@symbol These are two abbreviations that we use below to check symbols and accents in  
\tc@check@accent TS1. Only there to save some space, e.g., we can then write

```
\DeclareTextCommandDefault{\textcurrency}{\tc@check@symbol3\textcurrency}
```

to ensure that \textcurrency is only typeset if the current font has a TS1 subset id of less than 3. Otherwise \tc@error is called telling the user that for this font family \textcurrency is not available.

```
1531 \def\tc@check@symbol{\CheckEncodingSubset\UseTextSymbol{TS1}\tc@subst}
1532 \def\tc@check@accent{\CheckEncodingSubset\UseTextAccent{TS1}\tc@error}
```



We start with the commands that are “safe” and which can be unconditionally set up, first the accents...

```

1533 \DeclareTextAccentDefault{\capitalcedilla}{TS1}
1534 \DeclareTextAccentDefault{\capitalogonek}{TS1}
1535 \DeclareTextAccentDefault{\capitalgrave}{TS1}
1536 \DeclareTextAccentDefault{\capitalacute}{TS1}
1537 \DeclareTextAccentDefault{\capitalcircumflex}{TS1}
1538 \DeclareTextAccentDefault{\capitaltilde}{TS1}
1539 \DeclareTextAccentDefault{\capitaldieresis}{TS1}
1540 \DeclareTextAccentDefault{\capitalhungarumlaut}{TS1}
1541 \DeclareTextAccentDefault{\capitalring}{TS1}
1542 \DeclareTextAccentDefault{\capitalcaron}{TS1}
1543 \DeclareTextAccentDefault{\capitalbreve}{TS1}
1544 \DeclareTextAccentDefault{\capitalmacron}{TS1}
1545 \DeclareTextAccentDefault{\capitaldotaccent}{TS1}

```

...and then the other glyphs.

```

1546 \DeclareTextSymbolDefault{\textcapitalcompwordmark}{TS1}
1547 \DeclareTextSymbolDefault{\textascendercompwordmark}{TS1}
1548 \DeclareTextSymbolDefault{\textquotestraightbase}{TS1}
1549 \DeclareTextSymbolDefault{\textquotestraightdblbase}{TS1}
1550 \DeclareTextSymbolDefault{\texttwelveudash}{TS1}
1551 \DeclareTextSymbolDefault{\textthreequartersemdash}{TS1}
1552 \DeclareTextSymbolDefault{\textdollar}{TS1}
1553 \DeclareTextSymbolDefault{\textquotesingle}{TS1}
1554 \DeclareTextSymbolDefault{\textasteriskcentered}{TS1}
1555 \DeclareTextSymbolDefault{\textfractionsolidus}{TS1}
1556 \DeclareTextSymbolDefault{\textminus}{TS1}
1557 \DeclareTextSymbolDefault{\textlbrackdbl}{TS1}
1558 \DeclareTextSymbolDefault{\textrbrackdbl}{TS1}
1559 \DeclareTextSymbolDefault{\textasciigrave}{TS1}
1560 \DeclareTextSymbolDefault{\texttildelow}{TS1}
1561 \DeclareTextSymbolDefault{\textasciibreve}{TS1}
1562 \DeclareTextSymbolDefault{\textasciicaron}{TS1}
1563 \DeclareTextSymbolDefault{\textgravedbl}{TS1}
1564 \DeclareTextSymbolDefault{\textacutedbl}{TS1}
1565 \DeclareTextSymbolDefault{\textdagger}{TS1}
1566 \DeclareTextSymbolDefault{\textdaggerdbl}{TS1}
1567 \DeclareTextSymbolDefault{\textbardbl}{TS1}
1568 \DeclareTextSymbolDefault{\textperthousand}{TS1}
1569 \DeclareTextSymbolDefault{\textbullet}{TS1}
1570 \DeclareTextSymbolDefault{\textcelsius}{TS1}
1571 \DeclareTextSymbolDefault{\textflorin}{TS1}
1572 \DeclareTextSymbolDefault{\texttrademark}{TS1}
1573 \DeclareTextSymbolDefault{\textcent}{TS1}
1574 \DeclareTextSymbolDefault{\textsterling}{TS1}
1575 \DeclareTextSymbolDefault{\textyen}{TS1}
1576 \DeclareTextSymbolDefault{\textbrokenbar}{TS1}
1577 \DeclareTextSymbolDefault{\textsection}{TS1}
1578 \DeclareTextSymbolDefault{\textasciidieresis}{TS1}

```



```

1579 \DeclareTextSymbolDefault{\textcopyright}{TS1}
1580 \DeclareTextSymbolDefault{\textordfeminine}{TS1}
1581 \DeclareTextSymbolDefault{\textlnot}{TS1}
1582 \DeclareTextSymbolDefault{\textregistered}{TS1}
1583 \DeclareTextSymbolDefault{\textasciimacron}{TS1}
1584 \DeclareTextSymbolDefault{\textdegree}{TS1}
1585 \DeclareTextSymbolDefault{\textpm}{TS1}
1586 \DeclareTextSymbolDefault{\texttwosuperior}{TS1}
1587 \DeclareTextSymbolDefault{\textthreesuperior}{TS1}
1588 \DeclareTextSymbolDefault{\textasciiacute}{TS1}
1589 \DeclareTextSymbolDefault{\textmu}{TS1}
1590 \DeclareTextSymbolDefault{\textparagraph}{TS1}
1591 \DeclareTextSymbolDefault{\textperiodcentered}{TS1}
1592 \DeclareTextSymbolDefault{\textonesuperior}{TS1}
1593 \DeclareTextSymbolDefault{\textordmasculine}{TS1}
1594 \DeclareTextSymbolDefault{\textonequarter}{TS1}
1595 \DeclareTextSymbolDefault{\textonehalf}{TS1}
1596 \DeclareTextSymbolDefault{\textthreequarters}{TS1}
1597 \DeclareTextSymbolDefault{\texttimes}{TS1}
1598 \DeclareTextSymbolDefault{\textdiv}{TS1}

```

The `\texteuro` is only available for subsets with id 4 or less. Otherwise we fake the glyph using `\tc@fake@euro`

```

1599 \DeclareTextCommandDefault{\texteuro}
1600   {\CheckEncodingSubset\UseTextSymbol{TS1}\tc@fake@euro5\texteuro}

```

The `\textohm` is only available for subsets with id 3 or less. Otherwise we produce an error.

```

1601 \DeclareTextCommandDefault{\textohm}{\tc@check@symbol4\textohm}

```

The `\textestimated` and `\textcurrency` are only provided for fonts with subset encoding with id 2 or less.

```

1602 \DeclareTextCommandDefault{\textestimated}%
1603   {\tc@check@symbol3\textestimated}
1604 \DeclareTextCommandDefault{\textcurrency}%
1605   {\tc@check@symbol3\textcurrency}

```

Nearly all of the remaining glyphs are provided only with fonts with id 1 or 0, i.e., are essentially complete.

```

1606 \DeclareTextCommandDefault{\capitaltie}%
1607   {\tc@check@accent2\capitaltie}
1608 \DeclareTextCommandDefault{\newtie}%
1609   {\tc@check@accent2\newtie}
1610 \DeclareTextCommandDefault{\capitalnewtie}%
1611   {\tc@check@accent2\capitalnewtie}
1612 \DeclareTextCommandDefault{\textleftarrow}%
1613   {\tc@check@symbol2\textleftarrow}
1614 \DeclareTextCommandDefault{\textrightarrow}%
1615   {\tc@check@symbol2\textrightarrow}
1616 \DeclareTextCommandDefault{\textblank}%
1617   {\tc@check@symbol2\textblank}

```



```

1618 \DeclareTextCommandDefault{\textdblhyphen}%
1619     {\tc@check@symbol2\textdblhyphen}
1620 \DeclareTextCommandDefault{\textzerooldstyle}%
1621     {\tc@check@symbol2\textzerooldstyle}
1622 \DeclareTextCommandDefault{\textoneoldstyle}%
1623     {\tc@check@symbol2\textoneoldstyle}
1624 \DeclareTextCommandDefault{\texttwooldstyle}%
1625     {\tc@check@symbol2\texttwooldstyle}
1626 \DeclareTextCommandDefault{\textthreeoldstyle}%
1627     {\tc@check@symbol2\textthreeoldstyle}
1628 \DeclareTextCommandDefault{\textfouroldstyle}%
1629     {\tc@check@symbol2\textfouroldstyle}
1630 \DeclareTextCommandDefault{\textfiveoldstyle}%
1631     {\tc@check@symbol2\textfiveoldstyle}
1632 \DeclareTextCommandDefault{\textsixoldstyle}%
1633     {\tc@check@symbol2\textsixoldstyle}
1634 \DeclareTextCommandDefault{\textsevenoldstyle}%
1635     {\tc@check@symbol2\textsevenoldstyle}
1636 \DeclareTextCommandDefault{\texteightoldstyle}%
1637     {\tc@check@symbol2\texteightoldstyle}
1638 \DeclareTextCommandDefault{\textnineoldstyle}%
1639     {\tc@check@symbol2\textnineoldstyle}
1640 \DeclareTextCommandDefault{\textlangle}%
1641     {\tc@check@symbol2\textlangle}
1642 \DeclareTextCommandDefault{\textrightangle}%
1643     {\tc@check@symbol2\textrightangle}
1644 \DeclareTextCommandDefault{\textmho}%
1645     {\tc@check@symbol2\textmho}
1646 \DeclareTextCommandDefault{\textbigcircle}%
1647     {\tc@check@symbol2\textbigcircle}
1648 \DeclareTextCommandDefault{\textuparrow}%
1649     {\tc@check@symbol2\textuparrow}
1650 \DeclareTextCommandDefault{\textdownarrow}%
1651     {\tc@check@symbol2\textdownarrow}
1652 \DeclareTextCommandDefault{\textborn}%
1653     {\tc@check@symbol2\textborn}
1654 \DeclareTextCommandDefault{\textdivorced}%
1655     {\tc@check@symbol2\textdivorced}
1656 \DeclareTextCommandDefault{\textdied}%
1657     {\tc@check@symbol2\textdied}
1658 \DeclareTextCommandDefault{\textleaf}%
1659     {\tc@check@symbol2\textleaf}
1660 \DeclareTextCommandDefault{\textmarried}%
1661     {\tc@check@symbol2\textmarried}
1662 \DeclareTextCommandDefault{\textmusicalnote}%
1663     {\tc@check@symbol2\textmusicalnote}
1664 \DeclareTextCommandDefault{\textdblhyphenchar}%
1665     {\tc@check@symbol2\textdblhyphenchar}
1666 \DeclareTextCommandDefault{\textdollaroldstyle}%
1667     {\tc@check@symbol2\textdollaroldstyle}

```



```

1668 \DeclareTextCommandDefault{\textcentoldstyle}%
1669     {\tc@check@symbol2\textcentoldstyle}
1670 \DeclareTextCommandDefault{\textcolonmonetary}%
1671     {\tc@check@symbol2\textcolonmonetary}
1672 \DeclareTextCommandDefault{\textwon}%
1673     {\tc@check@symbol2\textwon}
1674 \DeclareTextCommandDefault{\textnaira}%
1675     {\tc@check@symbol2\textnaira}
1676 \DeclareTextCommandDefault{\textguarani}%
1677     {\tc@check@symbol2\textguarani}
1678 \DeclareTextCommandDefault{\textpeso}%
1679     {\tc@check@symbol2\textpeso}
1680 \DeclareTextCommandDefault{\textlira}%
1681     {\tc@check@symbol2\textlira}
1682 \DeclareTextCommandDefault{\textrecipe}%
1683     {\tc@check@symbol2\textrecipe}
1684 \DeclareTextCommandDefault{\textinterrobang}%
1685     {\tc@check@symbol2\textinterrobang}
1686 \DeclareTextCommandDefault{\textinterrobangdown}%
1687     {\tc@check@symbol2\textinterrobangdown}
1688 \DeclareTextCommandDefault{\textdong}%
1689     {\tc@check@symbol2\textdong}
1690 \DeclareTextCommandDefault{\textpertenthousand}%
1691     {\tc@check@symbol2\textpertenthousand}
1692 \DeclareTextCommandDefault{\textpilcrow}%
1693     {\tc@check@symbol2\textpilcrow}
1694 \DeclareTextCommandDefault{\textbaht}%
1695     {\tc@check@symbol2\textbaht}
1696 \DeclareTextCommandDefault{\textnumero}%
1697     {\tc@check@symbol2\textnumero}
1698 \DeclareTextCommandDefault{\textdiscount}%
1699     {\tc@check@symbol2\textdiscount}
1700 \DeclareTextCommandDefault{\textopenbullet}%
1701     {\tc@check@symbol2\textopenbullet}
1702 \DeclareTextCommandDefault{\textservicemark}%
1703     {\tc@check@symbol2\textservicemark}
1704 \DeclareTextCommandDefault{\textlquill}%
1705     {\tc@check@symbol2\textlquill}
1706 \DeclareTextCommandDefault{\textrquill}%
1707     {\tc@check@symbol2\textrquill}
1708 \DeclareTextCommandDefault{\textcopyleft}%
1709     {\tc@check@symbol2\textcopyleft}
1710 \DeclareTextCommandDefault{\textcircledP}%
1711     {\tc@check@symbol2\textcircledP}
1712 \DeclareTextCommandDefault{\textreferencemark}%
1713     {\tc@check@symbol2\textreferencemark}
1714 \DeclareTextCommandDefault{\textsurd}%
1715     {\tc@check@symbol2\textsurd}

```

The `\textcircled` and `\t` are handled specially, unless the current font has a



subset id of 0 (i.e. full TS1) we pick the symbols up from the the math font encodings, i.e., the third argument to `\CheckEncodingSubset` uses `\UseTextAccent` to get them from there.

```
1716 \DeclareTextCommandDefault{\textcircled}
1717   {\CheckEncodingSubset\UseTextAccent{TS1}%
1718    {\UseTextAccent{OMS}}1\textcircled}
1719 \DeclareTextCommandDefault{\t}
1720   {\CheckEncodingSubset\UseTextAccent{TS1}%
1721    {\UseTextAccent{OML}}1\t}
```

Finally input the encoding-specific definitions for TS1 thus making the top-level definitions optimised for this encoding (and not for the default encoding, see section 20.2).

```
1722 \input{ts1enc.def}
```

Now having the new glyphs available we also want to make sure that they are used. For most cases this will automatically happen but for some glyphs there are inferior definitions already known to L<sup>A</sup>T<sub>E</sub>X which will prevent the usage of the TS1 versions (see section 20.1 above). So we better get rid of them:

```
1723 \UndeclareTextCommand{\textsterling}{OT1}
1724 \UndeclareTextCommand{\textdollar} {OT1}
```

Similar declarations should probably be made for other encodings like OT4 if they are in use.

```
1725 %\UndeclareTextCommand{\textsterling}{OT4}
1726 %\UndeclareTextCommand{\textdollar} {OT4}
```

From the T1 encoding there are two candidates for removal: `%0` and `%00` since these are both constructed from `%` followed by a tiny ‘o’ rather than being a single glyph. The problem with this approach is that in PostScript fonts this small zero is usually not available resulting in `%■` rather than `%0` while the real glyph (at least for `\textperthousand`) is available in the PostScript version of TS1. So for the moment we compromise by removing the T1 declaration for `\textperthousand` but keeping the one for `\textpertenthousand`. This will have the effect that with Computer Modern fonts everything will come out (although `%0` and `%00` are not taken from the same physical font) and with PostScript fonts `%0` will come out correctly while `%00` will most likely look like `%■` — which is probably an improvement over just getting a single ‘■’ to indicate a completely missing glyph, which would happen if we also ‘undeclared’ `\textpertenthousand`.

```
1727 \UndeclareTextCommand{\textperthousand}{T1}
1728 %\UndeclareTextCommand{\textpertenthousand}{T1}
```

## 2.2.1 Supporting oldstyle digits

```
1729 \DeclareRobustCommand\oldstylenums[1]{%
1730   \begingroup
1731   \ifmmode
1732     \mathgroup\symletters #1%
1733   \else
```



```

1734 \CheckEncodingSubset\@use@text@encoding{TS1}%
1735 {\PackageWarning{textcomp}%
1736 {Oldstyle digits unavailable for
1737 family \f@family.\MessageBreak
1738 Lining digits used instead}}%
1739 \tw@{#1}%
1740 \fi
1741 \endgroup
1742 }

```

### 2.2.2 Subset encoding defaults

For many font families commonly used in the T<sub>E</sub>X world we provide the subset encoding data here. Users can add additional font families in the file `textcomp.cfg` if they own other fonts.

However, if the option “forced” was given then all subset encoding specifications are ignored, so there is no point in setting any of them up:

```
1743 \iftc@forced \else
```

Computer modern based fonts (e.g., CM, CM-Bright, Concrete):

```

1744 \DeclareEncodingSubset{TS1}{cmr}      {0}
1745 \DeclareEncodingSubset{TS1}{cmss}     {0}
1746 \DeclareEncodingSubset{TS1}{cmtt}     {0}
1747 \DeclareEncodingSubset{TS1}{cmvtt}    {0}
1748 \DeclareEncodingSubset{TS1}{cmbr}     {0}
1749 \DeclareEncodingSubset{TS1}{cmtl}     {0}
1750 \DeclareEncodingSubset{TS1}{ccr}      {0}

```

PSNFSS fonts:

```

1751 \DeclareEncodingSubset{TS1}{ptm}      {4}
1752 \DeclareEncodingSubset{TS1}{pcr}      {4}
1753 \DeclareEncodingSubset{TS1}{phv}      {4}
1754 \DeclareEncodingSubset{TS1}{ppl}      {3}
1755 \DeclareEncodingSubset{TS1}{pag}      {4}
1756 \DeclareEncodingSubset{TS1}{pbk}      {4}
1757 \DeclareEncodingSubset{TS1}{pnc}      {4}
1758 \DeclareEncodingSubset{TS1}{pzc}      {4}
1759 \DeclareEncodingSubset{TS1}{bch}      {4}
1760 \DeclareEncodingSubset{TS1}{put}      {5}

```

Other CTAN fonts (probably not complete):

```

1761 \DeclareEncodingSubset{TS1}{uag}      {5}
1762 \DeclareEncodingSubset{TS1}{ugq}      {5}
1763 \DeclareEncodingSubset{TS1}{ul8}      {4}
1764 \DeclareEncodingSubset{TS1}{ul9}      {4} % (LuxiSans, one day)
1765 \DeclareEncodingSubset{TS1}{augie}    {5}
1766 \DeclareEncodingSubset{TS1}{dayrom}   {3}
1767 \DeclareEncodingSubset{TS1}{dayroms}  {3}
1768 \DeclareEncodingSubset{TS1}{pxr}      {0}
1769 \DeclareEncodingSubset{TS1}{pxss}     {0}
1770 \DeclareEncodingSubset{TS1}{pxtt}     {0}

```



```

1771 \DeclareEncodingSubset{TS1}{txr}      {0}
1772 \DeclareEncodingSubset{TS1}{txss}     {0}
1773 \DeclareEncodingSubset{TS1}{txtt}     {0}

```

Latin Modern and TeX Gyre:

```

1774 \DeclareEncodingSubset{TS1}{lmr}      {0}
1775 \DeclareEncodingSubset{TS1}{lmdh}     {0}
1776 \DeclareEncodingSubset{TS1}{lmss}     {0}
1777 \DeclareEncodingSubset{TS1}{lmssq}    {0}
1778 \DeclareEncodingSubset{TS1}{lmvtt}    {0}
1779 \DeclareEncodingSubset{TS1}{lmtt}     {0}

1780 \DeclareEncodingSubset{TS1}{qhv}      {0}
1781 \DeclareEncodingSubset{TS1}{qag}      {0}
1782 \DeclareEncodingSubset{TS1}{qbk}      {0}
1783 \DeclareEncodingSubset{TS1}{qcr}      {0}
1784 \DeclareEncodingSubset{TS1}{qcs}      {0}
1785 \DeclareEncodingSubset{TS1}{qpl}      {0}
1786 \DeclareEncodingSubset{TS1}{qtm}      {0}
1787 \DeclareEncodingSubset{TS1}{qzc}      {0}
1788 \DeclareEncodingSubset{TS1}{qhvc}     {0}

```

Fourier-GUTenberg:

```

1789 \DeclareEncodingSubset{TS1}{futs}     {4}
1790 \DeclareEncodingSubset{TS1}{futex}    {4}
1791 \DeclareEncodingSubset{TS1}{futj}     {4}

```

Y&Y's Lucida Bright

```

1792 \DeclareEncodingSubset{TS1}{hlh}      {3}
1793 \DeclareEncodingSubset{TS1}{hls}      {3}
1794 \DeclareEncodingSubset{TS1}{hlst}     {3}

```

The remaining settings for Lucida are conservative: the following fonts contain the `\textohm` character but not the `\texteuro`, i.e., belong to neither subset 4 nor subset 3. If you want to use the `\textohm` with these fonts copy these definition to `textcomp.cfg` and change the subset to 3. However in that case make sure that you do not use the `\texteuro`.

```

1795 \DeclareEncodingSubset{TS1}{hlct}     {5}
1796 \DeclareEncodingSubset{TS1}{hlx}      {5}
1797 \DeclareEncodingSubset{TS1}{hlce}     {5}
1798 \DeclareEncodingSubset{TS1}{hlcn}     {5}
1799 \DeclareEncodingSubset{TS1}{hlcw}     {5}
1800 \DeclareEncodingSubset{TS1}{hlcf}     {5}

```

Other commercial families...

```

1801 \DeclareEncodingSubset{TS1}{pplx}     {3}
1802 \DeclareEncodingSubset{TS1}{pplj}     {3}
1803 \DeclareEncodingSubset{TS1}{ptmx}     {4}
1804 \DeclareEncodingSubset{TS1}{ptmj}     {4}

```

If the file `textcomp.cfg` exists it will be loaded at this point. This allows to define further subset encodings for font families not covered by default.



```
1805 \InputIfFileExists{textcomp.cfg}  
1806   {\PackageInfo{textcomp}{Local configuration file used}}{}  
1807 \fi  
1808 \end{TS1sty}
```