

The **telprint** package

Heiko Oberdiek
<oberdiek@uni-freiburg.de>

2007/09/09 v1.9

Abstract

Package **telprint** provides `\telprint` for formatting German phone numbers.

Contents

1	Documentation	2
1.1	Introduction	2
1.2	Short overview in English	2
1.2.1	Configuration	2
1.3	Documentation in German	2
2	Implementation	3
2.1	Reload check and package identification	3
2.2	Catcodes	4
2.3	Package macros	5
3	Test	7
3.1	Catcode checks for loading	7
4	Installation	8
4.1	Download	8
4.2	Bundle installation	9
4.3	Package installation	9
4.4	Refresh file name databases	9
4.5	Some details for the interested	9
5	History	10
	[1996/11/28 v1.0]	10
	[1997/09/16 v1.1]	10
	[1997/10/16 v1.2]	10
	[1997/12/09 v1.3]	10
	[2004/11/02 v1.4]	10
	[2005/09/30 v1.5]	10
	[2006/02/12 v1.6]	10
	[2006/08/26 v1.7]	10
	[2007/04/11 v1.8]	11
	[2007/09/09 v1.9]	11
6	Index	11

1 Documentation

1.1 Introduction

This is a very old package that I have written to format phone numbers. It follows German conventions and the documentation is mainly in German.

1.2 Short overview in English

L^AT_EX:

```
\usepackage{telprint}
\telprint{123/456-789}
```

plain-T_EX:

```
\input telprint.sty
\telprint{123/456-789}
```

\telprint **\telprint{...}** formats the explicitly given number. Digits, spaces and some special characters ('+', '/', '-', '(', ')', '~', ' ') are supported. Numbers are divided into groups of two digits from the right. Examples:

```
\telprint{0761/12345}      ==> 07\,61/1\,23\,45
\telprint{01234/567-89}    ==> 0\,12\,34/5\,67\leavevmode\hbox{-}89
\telprint{+49 (6221) 297}  ==> +49~(62\,21)~2\,97
```

1.2.1 Configuration

The output of the symbols can be configured by **\telhyphen**, **\telslash**, **\telleftparen**, **\telrightparen**, **\telplus**, **\teltilde**. Example:

```
\telslash{\,/\/,}\ \telprint{12/34} ==> 12\,/\/,34
```

\telspace **\telspace** configures the space between digit groups.
\telnumber **\telnumber** only formats a number in digit groups; special characters are not recognized.

1.3 Documentation in German

\telprint • **telprint#1**
Der eigentliche Anwenderbefehl zur formatierten Ausgabe von Telefonnummern. Diese dürfen dabei nur als Zahlen angegeben werden (, da sie tokenweise analysiert werden). Als Trenn- oder Sonderzeichen werden unterstützt: '+', '/', '-', '(', ')', '~', ' '. Einfache Leerzeichen werden erkannt und durch Tilden ersetzt, um Trennungen in der Telefonnummer zu verhindern. (Man beachte aus gleichem Grunde die **\hbox** bei '-'). Beispiele:

```
\telprint{0761/12345}      ==> 07\,61/1\,23\,45
\telprint{01234/567-89}    ==> 0\,12\,34/5\,67\leavevmode\hbox{-}89
\telprint{+49 (6221) 297}  ==> +49~(62\,21)~2\,97
```

Der Rest enthält eher Technisches:

\telspace • **\telspace#1**
Mit diesem Befehl wird der Abstand zwischen den Zifferngruppen angegeben (Default: \,). (Durch **\telspace{}** kann dieser zusätzliche Abstand abgestellt werden.)

<code>\telhyphen</code>	<ul style="list-style-type: none"> • <code>\telhyphen#1</code> Dieser Befehl gibt die Art des Bindestriches, wie er ausgegeben werden soll. In der Eingabe darf jedoch nur der einfache Bindestrich stehen: <code>\telprint{123-45}</code>, jedoch NIE <code>\telprint{123--45}</code>! Kopka-Bindestrich-Fans geben an: <code>\telhyphen{\leavevmode\hbox{--}}</code>
<code>\telslash</code>	<ul style="list-style-type: none"> • <code>\telslash#1, \telleftparen#1, \telrightparen#1, \telplus#1, \teltilde</code> Diese Befehle konfigurieren die Zeichen <code>'/'</code>, <code>'('</code>, <code>')'</code>, <code>'+'</code> und <code>'~'</code>. Sie funktionieren analog zu <code>\telhyphen</code>. • <code>\telnumber#1</code> Richtung interner Befehl: Er dient dazu, eine Zifferngruppe in Zweiergruppen auszugeben. Die einzelnen Zahlen werden im Tokenregister <code>\TELtoks</code> gespeichert. Abwechselnd werden dabei zwischen zwei Token (Zahlen) <code>\TELx</code> bzw. <code>\TELy</code> eingefügt, abhängig von dem wechselnden Wert von <code>\TELswitch</code>. Zum Schluss kann dann einfach festgestellt werden ob die Nummer nun eine geradzahlige oder ungeradzahlige Zahl von Ziffern aufwies. Dem entsprechend wird <code>\TELx</code> mit dem Zusatzabstand belegt und <code>\TELy</code> leer definiert oder umgekehrt.) • <code>\TEL...</code> interne Befehle, Technisches: <code>\TELsplit</code> dient zur Aufteilung einer zusammengesetzten Telefonnummer (Vorwahl, Hauptnummer, Nebenstelle). In dieser Implementation werden als Trennzeichen nur <code>'/'</code> und <code>'-'</code> erkannt. Die einzelnen Bestandteile wie Vorwahl werden dann dem Befehl <code>\telnumber</code> zur Formatierung uebergeben. • Die Erkennung von einfachen Leerzeichen ist um einiges schwieriger: Die Tokenentrennung ueber Parameter <code>#1#2</code> funktioniert nicht für einfache Leerzeichen, da TeX sie <i>niemals</i> als eigenständige Argumente behandelt! (The TeXbook, Chapter 20, p. 201) (Anmerkung am Rande: Deshalb funktionieren die entsprechenden Tokenmakros auf S. 149 des Buches „Einführung in TeX“ von N. Schwarz (3. Aufl.) nicht, wenn im Tokenregister als erstes ein einfaches Leerzeichen steht!)
<code>\telleftparen</code>	
<code>\telrightparen</code>	
<code>\telplus</code>	
<code>\teltilde</code>	
<code>\telnumber</code>	

2 Implementation

```
1 (*package)
```

2.1 Reload check and package identification

Reload check, especially if the package is not used with L^AT_EX.

```
2 \begingroup
3 \catcode44 12 % ,
4 \catcode45 12 % -
5 \catcode46 12 % .
6 \catcode58 12 % :
7 \catcode64 11 % @
8 \expandafter\let\expandafter\x\csname ver@telprint.sty\endcsname
9 \ifcase 0%
10 \ifx\x\relax % plain
11 \else
12 \ifx\x\empty % LaTeX
13 \else
14 1%
15 \fi
16 \fi
17 \else
18 \catcode35 6 % #
19 \catcode123 1 % {
20 \catcode125 2 % }
```

```

21 \expandafter\ifx\csname PackageInfo\endcsname\relax
22 \def\x#1#2{%
23 \immediate\write-1{Package #1 Info: #2.}%
24 }%
25 \else
26 \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
27 \fi
28 \x{telprint}{The package is already loaded}%
29 \endgroup
30 \expandafter\endinput
31 \fi
32 \endgroup

```

Package identification:

```

33 \begingroup
34 \catcode35 6 % #
35 \catcode40 12 % (
36 \catcode41 12 % )
37 \catcode44 12 % ,
38 \catcode45 12 % -
39 \catcode46 12 % .
40 \catcode47 12 % /
41 \catcode58 12 % :
42 \catcode64 11 % @
43 \catcode123 1 % {
44 \catcode125 2 % }
45 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
46 \def\x#1#2#3[#4]{\endgroup
47 \immediate\write-1{Package: #3 #4}%
48 \xdef#1{#4}%
49 }%
50 \else
51 \def\x#1#2[#3]{\endgroup
52 #2[#{#3}]%
53 \ifx#1\relax
54 \xdef#1{#3}%
55 \fi
56 }%
57 \fi
58 \expandafter\x\csname ver@telprint.sty\endcsname
59 \ProvidesPackage{telprint}%
60 [2007/09/09 v1.9 Formatting of German phone numbers (HO)]

```

2.2 Catcodes

```

61 \begingroup
62 \catcode123 1 % {
63 \catcode125 2 % }
64 \def\x{\endgroup
65 \expandafter\edef\csname TELAtEnd\endcsname{%
66 \catcode35 \the\catcode35\relax
67 \catcode64 \the\catcode64\relax
68 \catcode123 \the\catcode123\relax
69 \catcode125 \the\catcode125\relax
70 }%
71 }%
72 \x
73 \catcode35 6 % #
74 \catcode64 11 % @
75 \catcode123 1 % {
76 \catcode125 2 % }
77 \def\TMP@EnsureCode#1#2{%
78 \edef\TELAtEnd{%

```

```

79     \TELAtEnd
80     \catcode#1 \the\catcode#1\relax
81 }%
82 \catcode#1 #2\relax
83 }
84 \TMP@EnsureCode{33}{12}% !
85 \TMP@EnsureCode{36}{3}% $
86 \TMP@EnsureCode{40}{12}% (
87 \TMP@EnsureCode{41}{12}% )
88 \TMP@EnsureCode{42}{12}% *
89 \TMP@EnsureCode{43}{12}% +
90 \TMP@EnsureCode{44}{12}% ,
91 \TMP@EnsureCode{45}{12}% -
92 \TMP@EnsureCode{46}{12}% .
93 \TMP@EnsureCode{47}{12}% /
94 \TMP@EnsureCode{61}{12}% =
95 \TMP@EnsureCode{126}{13}% ~ (active)

```

2.3 Package macros

```

96 \ifx\DeclareRobustCommand\UnDeFiNeD
97   \def\DeclareRobustCommand#1[1]{\def#1##1}%
98   \def\TELreset{\let\DeclareRobustCommand=\UnDeFiNeD}%
99   \input infwarerr.sty\relax
100   \@PackageInfo{telprint}{%
101     Macros are not robust!%
102   }%
103 \else
104   \let\TELreset=\relax
105 \fi

\telspace

106 \DeclareRobustCommand*\telspace[1]{\def\TELspace{#1}}
107 \telspace{ }\$, $\{ }

\telhyphen

108 \DeclareRobustCommand*\telhyphen[1]{\def\TELhyphen{#1}}
109 \telhyphen{\leavevmode\hbox{-}}% \hbox zur Verhinderung der Trennung

\telslash

110 \DeclareRobustCommand*\telslash[1]{\def\TELslash{#1}}
111 \telslash{/}%

\telleftparen

112 \DeclareRobustCommand*\telleftparen[1]{\def\TELleftparen{#1}}
113 \telleftparen{(}%

\telrightparen

114 \DeclareRobustCommand*\telrightparen[1]{\def\TELrightparen{#1}}
115 \telrightparen{)}%

\telplus

116 \DeclareRobustCommand*\telplus[1]{\def\TELplus{#1}}
117 \telplus{+}%

\teltilde

118 \DeclareRobustCommand*\teltilde[1]{\def\TELtilde{#1}}
119 \teltilde{~}%

\TELtoks

120 \newtoks\TELtoks

```

\TELnumber

```
121 \def\TELnumber#1#2\TELnumberEND{%
122   \begingroup
123   \def\0{#2}%
124   \expandafter\endgroup
125   \ifx\0\empty
126     \TELtoks=\expandafter{\the\TELtoks#1}%
127     \ifnum\TELswitch=0 %
128       \def\TELx{\TELSpace}\def\TELy{}%
129     \else
130       \def\TELx{}\def\TELy{\TELSpace}%
131     \fi
132     \the\TELtoks
133   \else
134     \ifnum\TELswitch=0 %
135       \TELtoks=\expandafter{\the\TELtoks#1\TELx}%
136       \def\TELswitch{1}%
137     \else
138       \TELtoks=\expandafter{\the\TELtoks#1\TELy}%
139       \def\TELswitch{0}%
140     \fi
141     \TELnumber#2\TELnumberEND
142   \fi
143 }
```

\telnumber

```
144 \DeclareRobustCommand*\telnumber}[1]{%
145   \TELtoks={}%
146   \def\TELswitch{0}%
147   \TELnumber#1{}\TELnumberEND
148 }
```

\TELsplit

```
149 \def\TELsplit{\futurelet\TELfuture\TELdosplit}
```

\TELdosplit

```
150 \def\TELdosplit#1#2\TELsplitEND
151 {%
152   \def\TELsp{ }%
153   \expandafter\ifx\TELsp\TELfuture
154     \let\TELfuture=\relax
155     \expandafter\telnumber\expandafter{\the\TELtoks}~%
156     \telprint{#1#2}% Das Leerzeichen kann nicht #1 sein!
157   \else
158     \def\TELfirst{#1}%
159     \ifx\TELfirst\empty
160       \expandafter\telnumber\expandafter{\the\TELtoks}%
161       \TELtoks={}%
162     \else\if-\TELfirst
163       \expandafter\telnumber\expandafter{\the\TELtoks}\TELhyphen
164       \telprint{#2}%
165     \else\if/\TELfirst
166       \expandafter\telnumber\expandafter{\the\TELtoks}\TELslash
167       \telprint{#2}%
168     \else\if(\TELfirst
169       \expandafter\telnumber\expandafter{\the\TELtoks}\TELleftparen
170       \telprint{#2}%
171     \else\if)\TELfirst
172       \expandafter\telnumber\expandafter{\the\TELtoks}\TELrightparen
173       \telprint{#2}%
174     \else\if+\TELfirst
175       \expandafter\telnumber\expandafter{\the\TELtoks}\TELplus
```

```

176     \telprint{#2}%
177   \else\def\TELtemp{~}\ifx\TELtemp\TELfirst
178     \expandafter\telnumber\expandafter{\the\TELToks}\TELtilde
179     \telprint{#2}%
180   \else
181     \TELToks=\expandafter{\the\TELToks#1}%
182     \TELSplit#2{}\TELSplitEND
183     \fi\fi\fi\fi\fi\fi\fi
184   \fi
185 }

\telprint

186 \DeclareRobustCommand*\telprint}[1]{%
187   \TELToks={}
188   \TELSplit#1{}\TELSplitEND
189 }

190 \TELreset\let\TELreset=\UnDeFiNeD
191 \TELAteEnd
192 </package>

```

3 Test

3.1 Catcode checks for loading

```

193 <test1>

194 \catcode'\{=1 %
195 \catcode'\}=2 %
196 \catcode'\#=6 %
197 \catcode'\@=11 %
198 \expandafter\ifx\csname count@\endcsname\relax
199   \countdef\count@=255 %
200 \fi
201 \expandafter\ifx\csname @gobble\endcsname\relax
202   \long\def\@gobble#1{}%
203 \fi
204 \expandafter\ifx\csname @firstofone\endcsname\relax
205   \long\def\@firstofone#1{#1}%
206 \fi
207 \expandafter\ifx\csname loop\endcsname\relax
208   \expandafter\@firstofone
209 \else
210   \expandafter\@gobble
211 \fi
212 {%
213   \def\loop#1\repeat{%
214     \def\body{#1}%
215     \iterate
216   }%
217   \def\iterate{%
218     \body
219     \let\next\iterate
220   \else
221     \let\next\relax
222   \fi
223   \next
224 }%
225 \let\repeat=\fi
226 }%
227 \def\RestoreCatcodes{}
228 \count@=0 %

```

```

229 \loop
230   \edef\RestoreCatcodes{%
231     \RestoreCatcodes
232     \catcode\the\count@=\the\catcode\count@\relax
233   }%
234   \ifnum\count@<255 %
235     \advance\count@ 1 %
236   \repeat
237
238 \def\RangeCatcodeInvalid#1#2{%
239   \count@=#1\relax
240   \loop
241     \catcode\count@=15 %
242     \ifnum\count@<#2\relax
243       \advance\count@ 1 %
244     \repeat
245 }
246 \expandafter\ifx\csname LoadCommand\endcsname\relax
247   \def\LoadCommand{\input telprint.sty\relax}%
248 \fi
249 \def\Test{%
250   \RangeCatcodeInvalid{0}{47}%
251   \RangeCatcodeInvalid{58}{64}%
252   \RangeCatcodeInvalid{91}{96}%
253   \RangeCatcodeInvalid{123}{255}%
254   \catcode'\@=12 %
255   \catcode'\=0 %
256   \catcode'\{=1 %
257   \catcode'\}=2 %
258   \catcode'\#=6 %
259   \catcode'\[=12 %
260   \catcode'\]=12 %
261   \catcode'\%=14 %
262   \catcode'\ =10 %
263   \catcode13=5 %
264   \LoadCommand
265   \RestoreCatcodes
266 }
267 \Test
268 \csname @@end\endcsname
269 \end
270 </test1>

```

4 Installation

4.1 Download

Package. This package is available on CTAN¹:

[CTAN:macros/latex/contrib/oberdiek/telprint.dtx](#) The source file.

[CTAN:macros/latex/contrib/oberdiek/telprint.pdf](#) Documentation.

Bundle. All the packages of the bundle ‘oberdiek’ are also available in a TDS compliant ZIP archive. There the packages are already unpacked and the documentation files are generated. The files and directories obey the TDS standard.

[CTAN:macros/latex/contrib/oberdiek/oberdiek.tds.zip](#)

TDS refers to the standard “A Directory Structure for T_EX Files” ([CTAN:tds/tds.pdf](#)). Directories with `texmf` in their name are usually organized this way.

¹<http://ftp.ctan.org/tex-archive/>

4.2 Bundle installation

Unpacking. Unpack the `oberdiek.tds.zip` in the TDS tree (also known as `texmf` tree) of your choice. Example (linux):

```
unzip oberdiek.tds.zip -d ~/texmf
```

Script installation. Check the directory `TDS:scripts/oberdiek/` for scripts that need further installation steps. Package `attachfile2` comes with the Perl script `pdfatfi.pl` that should be installed in such a way that it can be called as `pdfatfi`. Example (linux):

```
chmod +x scripts/oberdiek/pdfatfi.pl
cp scripts/oberdiek/pdfatfi.pl /usr/local/bin/
```

4.3 Package installation

Unpacking. The `.dtx` file is a self-extracting `docstrip` archive. The files are extracted by running the `.dtx` through plain- $\mathrm{\TeX}$:

```
tex telprint.dtx
```

TDS. Now the different files must be moved into the different directories in your installation TDS tree (also known as `texmf` tree):

<code>telprint.sty</code>	\rightarrow <code>tex/generic/oberdiek/telprint.sty</code>
<code>telprint.pdf</code>	\rightarrow <code>doc/latex/oberdiek/telprint.pdf</code>
<code>test/telprint-test1.tex</code>	\rightarrow <code>doc/latex/oberdiek/test/telprint-test1.tex</code>
<code>telprint.dtx</code>	\rightarrow <code>source/latex/oberdiek/telprint.dtx</code>

If you have a `docstrip.cfg` that configures and enables `docstrip`'s TDS installing feature, then some files can already be in the right place, see the documentation of `docstrip`.

4.4 Refresh file name databases

If your $\mathrm{\TeX}$ distribution (te $\mathrm{\TeX}$, mik $\mathrm{\TeX}$, ...) relies on file name databases, you must refresh these. For example, te $\mathrm{\TeX}$ users run `texhash` or `mktextlsr`.

4.5 Some details for the interested

Attached source. The PDF documentation on CTAN also includes the `.dtx` source file. It can be extracted by AcrobatReader 6 or higher. Another option is `pdftk`, e.g. unpack the file into the current directory:

```
pdftk telprint.pdf unpack_files output .
```

Unpacking with $\mathrm{\LaTeX}$. The `.dtx` chooses its action depending on the format:

plain- $\mathrm{\TeX}$: Run `docstrip` and extract the files.

$\mathrm{\LaTeX}$: Generate the documentation.

If you insist on using $\mathrm{\LaTeX}$ for `docstrip` (really, `docstrip` does not need $\mathrm{\LaTeX}$), then inform the autodetect routine about your intention:

```
latex \let\install=y\input{telprint.dtx}
```

Do not forget to quote the argument according to the demands of your shell.

Generating the documentation. You can use both the `.dtx` or the `.drv` to generate the documentation. The process can be configured by the configuration file `ltxdoc.cfg`. For instance, put this line into this file, if you want to have A4 as paper format:

```
\PassOptionsToClass{a4paper}{article}
```

An example follows how to generate the documentation with pdfL^AT_EX:

```
pdflatex telprint.dtx
makeindex -s gind.ist telprint.idx
pdflatex telprint.dtx
makeindex -s gind.ist telprint.idx
pdflatex telprint.dtx
```

5 History

[1996/11/28 v1.0]

- Erste lauffähige Version.
- Nur `'` und `'/'` als zulässige Sonderzeichen.

[1997/09/16 v1.1]

- Dokumentation und Kommentare (Posting in `de.comp.text.tex`).
- Erweiterung um Sonderzeichen `'(, ')`, `'+'`, `'~'` und `' '`.
- Trennungsverhinderung am `'hyphen'`.

[1997/10/16 v1.2]

- Schutz vor wiederholtem Einlesen.
- Unter L^AT_EX 2_ε Nutzung des `\DeclareRobustCommand`-Features.

[1997/12/09 v1.3]

- Temporäre Variable eingespart.

[2004/11/02 v1.4]

- Fehler in der Dokumentation korrigiert.

[2005/09/30 v1.5]

- Konfigurierbare Symbole: `'/'`, `'(, ')`, `'+'` und `'~'`.

[2006/02/12 v1.6]

- LPPL 1.3.
- Kurze Übersicht in Englisch.
- CTAN.

[2006/08/26 v1.7]

- New DTX framework.

[2007/04/11 v1.8]

- Line ends sanitized.

[2007/09/09 v1.9]

- Catcode section added.
- Missing docstrip tag added.

6 Index

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

Symbols	\end 269
\# 196, 258	\endcsname 8, 21, 45,
\% 261	58, 65, 198, 201, 204, 207, 246, 268
\, 107	\endinput 30
\@ 197, 254	
\@PackageInfo 100	F
\@firstofone 205, 208	\futurelet 149
\@gobble 202, 210	
\[. 259	H
\] 255	\hbox 109
\{ 194, 256	
\} 195, 257	I
\] 260	\if 162, 165, 168, 171, 174
	\ifcase 9
Numbers	\ifnum 127, 134, 234, 242
\0 123, 125	\ifx . 10, 12, 21, 45, 53, 96, 125, 153,
	159, 177, 198, 201, 204, 207, 246
_ 262	\immediate 23, 47
	\input 99, 247
	\iterate 215, 217, 219
A	
\advance 235, 243	L
B	\leavevmode 109
\body 214, 218	\LoadCommand 247, 264
C	\loop 213, 229, 240
\catcode . . . 3, 4, 5, 6, 7, 18, 19, 20,	N
34, 35, 36, 37, 38, 39, 40, 41, 42,	\newtoks 120
43, 44, 62, 63, 66, 67, 68, 69, 73,	\next 219, 221, 223
74, 75, 76, 80, 82, 194, 195, 196,	
197, 232, 241, 254, 255, 256,	P
257, 258, 259, 260, 261, 262, 263	\PackageInfo 26
\count@ 199, 228,	\ProvidesPackage 59
232, 234, 235, 239, 241, 242, 243	
\countdef 199	R
\csname 8, 21, 45,	\RangeCatcodeInvalid
58, 65, 198, 201, 204, 207, 246, 268	238, 250, 251, 252, 253
D	\repeat 213, 225, 236, 244
\DeclareRobustCommand	\RestoreCatcodes . . 227, 230, 231, 265
96, 97, 98, 106, 108,	
110, 112, 114, 116, 118, 144, 186	T
E	\TELatEnd 78, 79, 191
\empty 12, 125, 159	\TELdosplit 149, <u>150</u>
	\TELfirst 158,
	159, 162, 165, 168, 171, 174, 177
	\TELfuture 149, 153, 154
	\TELhyphen 108, 163

<code>\telhyphen</code>	3 , 108	<code>\TELtemp</code>	177
<code>\TELleftparen</code>	112 , 169	<code>\TELtilde</code>	118 , 178
<code>\telleftparen</code>	3 , 112	<code>\teltilde</code>	3 , 118
<code>\TELnumber</code>	121 , 147	<code>\TELtoks</code>	120 , 126 , 132 , 135 ,
<code>\telnumber</code>	2 , 3 , 144 , 155 ,		138 , 145 , 155 , 160 , 161 , 163 ,
	160 , 163 , 166 , 169 , 172 , 175 , 178		166 , 169 , 172 , 175 , 178 , 181 , 187
<code>\TELnumberEND</code>	121 , 141 , 147	<code>\TELx</code>	128 , 130 , 135
<code>\TELplus</code>	116 , 175	<code>\TELy</code>	128 , 130 , 138
<code>\telplus</code>	3 , 116	<code>\Test</code>	249 , 267
<code>\telprint</code>	2 , 2 , 156 ,	<code>\the</code>	66 , 67 , 68 , 69 , 80 , 126 ,
	164 , 167 , 170 , 173 , 176 , 179 , 186		132 , 135 , 138 , 155 , 160 , 163 ,
<code>\TELreset</code>	98 , 104 , 190		166 , 169 , 172 , 175 , 178 , 181 , 232
<code>\TELrightparen</code>	114 , 172	<code>\TMP@EnsureCode</code>	77 , 84 , 85 ,
<code>\telrightparen</code>	3 , 114		86 , 87 , 88 , 89 , 90 , 91 , 92 , 93 , 94 , 95
<code>\TELslash</code>	110 , 166		
<code>\telslash</code>	3 , 110		
<code>\TELsp</code>	152 , 153		
<code>\TELspace</code>	106 , 128 , 130		
<code>\telspace</code>	2 , 2 , 106		
<code>\TELsplit</code>	149 , 182 , 188		
<code>\TELsplitEND</code>	150 , 182 , 188		
<code>\TELswitch</code> ...	127 , 134 , 136 , 139 , 146		

U

`\UnDeFiNeD`

W

`\write`

X

`\x` [8](#), [10](#), [12](#), [22](#), [26](#), [28](#), [46](#), [51](#), [58](#), [64](#), [72](#)