The telprint package

Heiko Oberdiek*

2016/05/16 v1.11

Abstract

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

Contents

*Please report any issues at https://github.com/ho-tex/oberdiek/issues

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

LaTeX:

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

plain TeX:

```
\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 zusaetzliche Abstand abgestellt werden.)

\telhyphen

- `\telhyphen#1`
  Dieser Befehl gibt die Art des Bindestriches, wie er ausgegeben werden soll. In der Eingabe darf jedoch nur der einfache Bindestrich stehen: `\telprint{123-45}`, jedoch NIE `\telprint{123--45}`! Kopka-Bindestrich-Fans geben an: `\telhyphen{\leavevmode\hbox{--}}`

\telslash
\telleftparen
\telrightparen
\telplus
\teltilde

- `\telslash#1`, `\telleftparen#1`, `\telrightparen#1`, `\telplus#1`, `\teltilde`
  Diese Befehle konfigurieren die Zeichen '/', '(', ')', '+' und '~'. Sie funktionieren analog zu `\telhyphen`.

\telnumber

- `\telnumber#1`
  Richtung interner Befehl: Er dient dazu, eine Zifferngruppe in Zweiergruppen auszugeben. Die einzelnen Zahlen werden im Tokenregister `\TELtoks` gespeichert. Abwechselnd werden dabei zwischen zwei Token (Zahlen) `\TELx` bzw. `\TELy` eingefuegt, abhängig von dem wechselnden Wert von `\TELswitch`. Zum Schluss kann dann einfach festgestellt werden ob die Nummer nun eine geradzahlige oder ungeradzahlige Zahl von Ziffern aufwies. Dem entsprechend wird `\TELx` mit dem Zusatzabstand belegt und `\TELy` leer definiert oder umgekehrt.)

- `\TEL...` interne Befehle, Technisches:
  `\TELsplit` dient zur Aufteilung einer zusammengesetzten Telefonnummer (Vorwahl, Hauptnummer, Nebenstelle). In dieser Implementation werden als Trennzeichen nur '/' und '-' erkannt. Die einzelnen Bestandteile wie Vorwahl werden dann dem Befehl `\telnumber` zur Formatierung uebergeben.

- Die Erkennung von einfachen Leerzeichen ist um einiges schwieriger: Die Tokentrennung ueber Parameter `#1#2` funktioniert nicht für einfache Leerzeichen, da TeX sie *niemals* 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!)

2 Implementation

1 ⟨*package⟩

2.1 Reload check and package identification

Reload check, especially if the package is not used with LaTeX.

```
2 \begingroup\catcode61\catcode48\catcode32=10\relax%
3   \catcode13=5 % ^^M
4   \endlinechar=13 %
5   \catcode35=6 % #
6   \catcode39=12 % '
7   \catcode44=12 % ,
8   \catcode45=12 % -
9   \catcode46=12 % .
10  \catcode58=12 % :
11  \catcode64=11 % @
12  \catcode123=1 % {
13  \catcode125=2 % }
14  \expandafter\let\expandafter\x\csname ver@telprint.sty\endcsname
15  \ifx\x\relax % plain-TeX, first loading
16  \else
```

```
    \def\empty{}%
    \ifx\x\empty % LaTeX, first loading,
      % variable is initialized, but \ProvidesPackage not yet seen
    \else
      \expandafter\ifx\csname PackageInfo\endcsname\relax
        \def\x#1#2{%
          \immediate\write-1{Package #1 Info: #2.}%
        }%
      \else
        \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
      \fi
      \x{telprint}{The package is already loaded}%
      \aftergroup\endinput
    \fi
  \fi
\endgroup%
```

Package identification:

```
\begingroup\catcode61\catcode48\catcode32=10\relax%
  \catcode13=5 % ^^M
  \endlinechar=13 %
  \catcode35=6 % #
  \catcode39=12 % '
  \catcode40=12 % (
  \catcode41=12 % )
  \catcode44=12 % ,
  \catcode45=12 % -
  \catcode46=12 % .
  \catcode47=12 % /
  \catcode58=12 % :
  \catcode64=11 % @
  \catcode91=12 % [
  \catcode93=12 % ]
  \catcode123=1 % {
  \catcode125=2 % }
  \expandafter\ifx\csname ProvidesPackage\endcsname\relax
    \def\x#1#2#3[#4]{\endgroup
      \immediate\write-1{Package: #3 #4}%
      \xdef#1{#4}%
    }%
  \else
    \def\x#1#2[#3]{\endgroup
      #2[{#3}]%
      \ifx#1\@undefined
        \xdef#1{#3}%
      \fi
      \ifx#1\relax
        \xdef#1{#3}%
      \fi
    }%
  \fi
\expandafter\x\csname ver@telprint.sty\endcsname
\ProvidesPackage{telprint}%
  [2016/05/16 v1.11 Format German phone numbers (HO)]%
```

2.2 Catcodes

```
\begingroup\catcode61\catcode48\catcode32=10\relax%
  \catcode13=5 % ^^M
```

```
  \endlinechar=13 %
  \catcode123=1 % {
  \catcode125=2 % }
  \catcode64=11 % @
  \def\x{\endgroup
    \expandafter\edef\csname TELAtEnd\endcsname{%
      \endlinechar=\the\endlinechar\relax
      \catcode13=\the\catcode13\relax
      \catcode32=\the\catcode32\relax
      \catcode35=\the\catcode35\relax
      \catcode61=\the\catcode61\relax
      \catcode64=\the\catcode64\relax
      \catcode123=\the\catcode123\relax
      \catcode125=\the\catcode125\relax
    }%
  }%
\x\catcode61\catcode48\catcode32=10\relax%
\catcode13=5 % ^^M
\endlinechar=13 %
\catcode35=6 % #
\catcode64=11 % @
\catcode123=1 % {
\catcode125=2 % }
\def\TMP@EnsureCode#1#2{%
  \edef\TELAtEnd{%
    \TELAtEnd
    \catcode#1=\the\catcode#1\relax
  }%
  \catcode#1=#2\relax
}
\TMP@EnsureCode{33}{12}% !
\TMP@EnsureCode{36}{3}% $
\TMP@EnsureCode{40}{12}% (
\TMP@EnsureCode{41}{12}% )
\TMP@EnsureCode{42}{12}% *
\TMP@EnsureCode{43}{12}% +
\TMP@EnsureCode{44}{12}% ,
\TMP@EnsureCode{45}{12}% -
\TMP@EnsureCode{46}{12}% .
\TMP@EnsureCode{47}{12}% /
\TMP@EnsureCode{91}{12}% [
\TMP@EnsureCode{93}{12}% ]
\TMP@EnsureCode{126}{13}% ~ (active)
\edef\TELAtEnd{\TELAtEnd\noexpand\endinput}
```

2.3 Package macros

```
\ifx\DeclareRobustCommand\UnDeFiNeD
  \def\DeclareRobustCommand*#1[1]{\def#1##1}%
  \def\TELreset{\let\DeclareRobustCommand=\UnDeFiNeD}%
  \input infwarerr.sty\relax
  \@PackageInfo{telprint}{%
    Macros are not robust!%
  }%
\else
  \let\TELreset=\relax
\fi
```

\telspace

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

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

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

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

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

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

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

\TELtoks
\newtoks\TELtoks

\TELnumber
\def\TELnumber#1#2\TELnumberEND{%
  \begingroup
  \def\0{#2}%
  \expandafter\endgroup
  \ifx\0\empty
    \TELtoks=\expandafter{\the\TELtoks#1}%
    \ifnum\TELswitch=0 %
      \def\TELx{\TELspace}\def\TELy{}%
    \else
      \def\TELx{}\def\TELy{\TELspace}%
    \fi
    \the\TELtoks
  \else
    \ifnum\TELswitch=0 %
      \TELtoks=\expandafter{\the\TELtoks#1\TELx}%
      \def\TELswitch{1}%
    \else
      \TELtoks=\expandafter{\the\TELtoks#1\TELy}%
      \def\TELswitch{0}%
    \fi
    \TELnumber#2\TELnumberEND
  \fi
}
```

\telnumber

```
\DeclareRobustCommand*{\telnumber}[1]{%
  \TELtoks={}%
  \def\TELswitch{0}%
  \TELnumber#1{}\TELnumberEND
}
```

\TELsplit

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

\TELdosplit

```
\def\TELdosplit#1#2\TELsplitEND
{%
  \def\TELsp{ }%
  \expandafter\ifx\TELsp\TELfuture
    \let\TELfuture=\relax
    \expandafter\telnumber\expandafter{\the\TELtoks}~%
    \telprint{#1#2}% Das Leerzeichen kann nicht #1 sein!
  \else
    \def\TELfirst{#1}%
    \ifx\TELfirst\empty
      \expandafter\telnumber\expandafter{\the\TELtoks}%
      \TELtoks={}%
    \else\if-\TELfirst
      \expandafter\telnumber\expandafter{\the\TELtoks}\TELhyphen
      \telprint{#2}%
    \else\if/\TELfirst
      \expandafter\telnumber\expandafter{\the\TELtoks}\TELslash
      \telprint{#2}%
    \else\if(\TELfirst
      \expandafter\telnumber\expandafter{\the\TELtoks}\TELleftparen
      \telprint{#2}%
    \else\if)\TELfirst
      \expandafter\telnumber\expandafter{\the\TELtoks}\TELrightparen
      \telprint{#2}%
    \else\if+\TELfirst
      \expandafter\telnumber\expandafter{\the\TELtoks}\TELplus
      \telprint{#2}%
    \else\def\TELtemp{~}\ifx\TELtemp\TELfirst
      \expandafter\telnumber\expandafter{\the\TELtoks}\TELtilde
      \telprint{#2}%
    \else
      \TELtoks=\expandafter{\the\TELtoks#1}%
      \TELsplit#2{}\TELsplitEND
    \fi\fi\fi\fi\fi\fi\fi
  \fi
}
```

\telprint

```
\DeclareRobustCommand*{\telprint}[1]{%
  \TELtoks={}%
  \TELsplit#1{}\TELsplitEND
}

\TELreset\let\TELreset=\UnDeFiNeD

\TELAtEnd%
```

⟨/package⟩

3 Installation

3.1 Download

Package. This package is available on CTAN[1]:

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:install/macros/latex/contrib/oberdiek.tds.zip

TDS refers to the standard "A Directory Structure for TEX Files" (CTAN:pkg/tds). Directories with `texmf` in their name are usually organized this way.

3.2 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
```

3.3 Package installation

Unpacking. The `.dtx` file is a self-extracting docstrip archive. The files are extracted by running the `.dtx` through plain 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):

```
telprint.sty → tex/generic/oberdiek/telprint.sty
telprint.pdf → doc/latex/oberdiek/telprint.pdf
telprint.dtx → source/latex/oberdiek/telprint.dtx
```

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.

3.4 Refresh file name databases

If your TEX distribution (TEX Live, MiKTEX, ...) relies on file name databases, you must refresh these. For example, TEX Live users run `texhash` or `mktexlsr`.

3.5 Some details for the interested

Unpacking with LATEX. The `.dtx` chooses its action depending on the format:

plain TEX: Run docstrip and extract the files.

LATEX: Generate the documentation.

[1] CTAN:pkg/telprint

If you insist on using LaTeX for docstrip (really, docstrip does not need 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 pdfLaTeX:

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

4 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 LaTeX 2_ε Nutzung des `\DeclareRobustCommand`-Features.

[1997/12/09 v1.3]

- Temporäre Variable eingespart.
- Posted in newsgroup `de.comp.text.tex`:
  "Re: Generisches Markup für Telefonnummern?"[2]

[2004/11/02 v1.4]

- Fehler in der Dokumentation korrigiert.

[2]Url: https://groups.google.com/group/de.comp.text.tex/msg/86b3a86140007309

[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.

[2008/08/11 v1.10]

- Code is not changed.
- URLs updated.

[2016/05/16 v1.11]

- Documentation updates.

5 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; plain numbers refer to the code lines where the entry is used.