The telprint package

Heiko Oberdiek
<oberdiek@uni-freiburg.de>

2008/08/11 v1.10

Abstract

Package telprint provides \telprint for formatting German phone numbers.

Contents

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

```
⟨*package⟩
```

2.1 Reload check and package identification

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

```
\begingroup
  \catcode44 12 % ,
  \catcode45 12 % -
  \catcode46 12 % .
  \catcode58 12 % :
  \catcode64 11 % @
  \expandafter\let\expandafter\x\csname ver@telprint.sty\endcsname
  \ifcase 0%
    \ifx\x\relax % plain
    \else
      \ifx\x\empty % LaTeX
      \else
        1%
      \fi
    \fi
  \else
    \catcode35 6 % #
    \catcode123 1 % {
    \catcode125 2 % }
```

```
    \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}%
    \endgroup
    \expandafter\endinput
  \fi
\endgroup
```

Package identification:

```
\begingroup
  \catcode35 6 % #
  \catcode40 12 % (
  \catcode41 12 % )
  \catcode44 12 % ,
  \catcode45 12 % -
  \catcode46 12 % .
  \catcode47 12 % /
  \catcode58 12 % :
  \catcode64 11 % @
  \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\relax
        \xdef#1{#3}%
      \fi
    }%
  \fi
\expandafter\x\csname ver@telprint.sty\endcsname
\ProvidesPackage{telprint}%
  [2008/08/11 v1.10 Formatting of German phone numbers (HO)]
```

2.2 Catcodes

```
\begingroup
  \catcode123 1 % {
  \catcode125 2 % }
  \def\x{\endgroup
    \expandafter\edef\csname TELAtEnd\endcsname{%
      \catcode35 \the\catcode35\relax
      \catcode64 \the\catcode64\relax
      \catcode123 \the\catcode123\relax
      \catcode125 \the\catcode125\relax
    }%
  }%
\x
\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{61}{12}% =
\TMP@EnsureCode{126}{13}% ~ (active)
```

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 Test

3.1 Catcode checks for loading

⟨*test1⟩

```
\catcode`\{=1 %
\catcode`\}=2 %
\catcode`\#=6 %
\catcode`\@=11 %
\expandafter\ifx\csname count@\endcsname\relax
  \countdef\count@=255 %
\fi
\expandafter\ifx\csname @gobble\endcsname\relax
  \long\def\@gobble#1{}%
\fi
\expandafter\ifx\csname @firstofone\endcsname\relax
  \long\def\@firstofone#1{#1}%
\fi
\expandafter\ifx\csname loop\endcsname\relax
  \expandafter\@firstofone
\else
  \expandafter\@gobble
\fi
{%
  \def\loop#1\repeat{%
    \def\body{#1}%
    \iterate
  }%
  \def\iterate{%
    \body
      \let\next\iterate
    \else
      \let\next\relax
    \fi
    \next
  }%
  \let\repeat=\fi
}%
\def\RestoreCatcodes{}
\count@=0 %
```

```
\loop
  \edef\RestoreCatcodes{%
    \RestoreCatcodes
    \catcode\the\count@=\the\catcode\count@\relax
  }%
\ifnum\count@<255 %
  \advance\count@ 1 %
\repeat

\def\RangeCatcodeInvalid#1#2{%
  \count@=#1\relax
  \loop
    \catcode\count@=15 %
  \ifnum\count@<#2\relax
    \advance\count@ 1 %
  \repeat
}
\expandafter\ifx\csname LoadCommand\endcsname\relax
  \def\LoadCommand{\input telprint.sty\relax}%
\fi
\def\Test{%
  \RangeCatcodeInvalid{0}{47}%
  \RangeCatcodeInvalid{58}{64}%
  \RangeCatcodeInvalid{91}{96}%
  \RangeCatcodeInvalid{123}{255}%
  \catcode`\@=12 %
  \catcode`\\=0 %
  \catcode`\{=1 %
  \catcode`\}=2 %
  \catcode`\#=6 %
  \catcode`\[=12 %
  \catcode`\]=12 %
  \catcode`\%=14 %
  \catcode`\ =10 %
  \catcode13=5 %
  \LoadCommand
  \RestoreCatcodes
}
\Test
\csname @@end\endcsname
\end
```

⟨/test1⟩

4 Installation

4.1 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:tds/tds.pdf). Directories with `texmf` in their name are usually organized this way.

[1] ftp://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-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
test/telprint-test1.tex → doc/latex/oberdiek/test/telprint-test1.tex
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.

4.4 Refresh file name databases

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

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

plain-TEX: Run docstrip and extract the files.

LATEX: Generate the documentation.

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
```

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 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.

[2005/09/30 v1.5]

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

[2006/02/12 v1.6]

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

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

[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.

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.