The soulutf8 package

Heiko Oberdiek*
<heiko.oberdiek at googlemail.com>

2016/05/16 v1.1

Abstract

This package extends package soul and adds some support for UTF-8. Namely the input encodings `utf8.def` from package inputenc and package ucs's `utf8x.def` are supported.

Contents

1 Documentation **2**
1.1 Patch . . . 2
1.2 Future . . . 2

2 Implementation **3**
2.1 Reload check and package identification . . . 3
2.2 Catcodes . . . 4
2.3 Loading packages . . . 5
2.3.1 plain TeX . . . 5
2.3.2 LaTeX . . . 6
2.3.3 ε-TeX . . . 6
2.4 Macro for redefinitions . . . 6
2.5 Redefinition of `\SOUL@eval` . . . 7
2.6 UTF-8 analysis . . . 10
2.6.1 Help strings . . . 10
2.6.2 Support for `utf8.def` . . . 10
2.6.3 Support for `utf8x.def` . . . 11
2.7 Actions for UTF-8 sequences . . . 12
2.7.1 Redefinition of `\SOUL@splittoken` . . . 12
2.8 Patches . . . 13

3 Test **16**
3.1 Catcode checks for loading . . . 16
3.2 LaTeXtests . . . 17
3.2.1 Driver definitions . . . 19
3.2.2 Test macros . . . 20
3.2.3 Testing . . . 23

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

4 Installation **24**
4.1 Download . . . 24
4.2 Bundle installation . . . 25
4.3 Package installation . . . 25
4.4 Refresh file name databases . . . 25
4.5 Some details for the interested . . . 25

5 Catalogue **26**

6 References **27**

7 History **27**
[2007/09/09 v1.0] . . . 27
[2016/05/16 v1.1] . . . 27

8 Index **27**

1 Documentation

This package soulutf8 does not have own options and does not define new user commands. Any option is passed to package soul [1] that is loaded first. Then some internal macros of soul are redefined to add support for UTF-8. The following input encodings are supported:

`utf8`	LaTeX base	`TDS:tex/latex/base/utf8.def` [3]
`utf8x`	Package ucs	`TDS:tex/latex/ucs/utf8x.def` [2]

UTF-8 byte sequences are added as token group to a word, even if these UTF-8 characters are some kind of hyphen or space. As exception the following three Unicode characters are handled specially:

Slot	Name	Action
`U+00A0`	NO-BREAK SPACE	like ~
`U+2013`	EN DASH	`--`
`U+2014`	EM DASH	`---`

1.1 Patch

Also package soulutf8 tries to patch package soul to improve its behaviour:

- A problem with additional levels of curly braces is fixed. As advantage more implicite kernings are detected. However, the result may be incompatible with the original behaviour of package soul because of these respected implicite kernings.
- ε-TeX , especially `\unexpanded` is supported. This allows a better protection of token groups (`\mbox{...}`, math, ...).

1.2 Future

Currently package soul does not seem to be maintained. Nevertheless if there will be a new version that adds support for UTF-8, then this package may become obsolete.

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@soulutf8.sty\endcsname
15  \ifx\x\relax % plain-TeX, first loading
16  \else
17    \def\empty{}%
18    \ifx\x\empty % LaTeX, first loading,
19      % variable is initialized, but \ProvidesPackage not yet seen
20    \else
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{soulutf8}{The package is already loaded}%
29      \aftergroup\endinput
30    \fi
31  \fi
32 \endgroup%
```

Package identification:

```
33 \begingroup\catcode61\catcode48\catcode32=10\relax%
34   \catcode13=5 % ^^M
35   \endlinechar=13 %
36   \catcode35=6 % #
37   \catcode39=12 % '
38   \catcode40=12 % (
39   \catcode41=12 % )
40   \catcode44=12 % ,
41   \catcode45=12 % -
42   \catcode46=12 % .
43   \catcode47=12 % /
44   \catcode58=12 % :
45   \catcode64=11 % @
46   \catcode91=12 % [
47   \catcode93=12 % ]
48   \catcode123=1 % {
49   \catcode125=2 % }
50   \expandafter\ifx\csname ProvidesPackage\endcsname\relax
51     \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@soulutf8.sty\endcsname
\ProvidesPackage{soulutf8}%
  [2016/05/16 v1.1 Permit use of UTF-8 characters in soul (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 SOuL@AtEnd\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\SOuL@AtEnd{%
    \SOuL@AtEnd
    \catcode#1=\the\catcode#1\relax
  }%
  \catcode#1=#2\relax
}
\TMP@EnsureCode{10}{12}% ^^J
\TMP@EnsureCode{33}{12}% !
\TMP@EnsureCode{34}{12}% "
\TMP@EnsureCode{36}{3}% $
\TMP@EnsureCode{39}{12}% '
\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{58}{12}% :
\TMP@EnsureCode{60}{12}% <
\TMP@EnsureCode{62}{12}% >
\TMP@EnsureCode{91}{12}% [
\TMP@EnsureCode{93}{12}% ]
\TMP@EnsureCode{94}{7}% ^
\TMP@EnsureCode{96}{12}% `
\TMP@EnsureCode{126}\active % ~
\TMP@EnsureCode{128}{12}% ^^80
\TMP@EnsureCode{147}{12}% ^^93
\TMP@EnsureCode{148}{12}% ^^94
\TMP@EnsureCode{160}{12}% ^^a0
\TMP@EnsureCode{194}{12}% ^^c2
\TMP@EnsureCode{226}{12}% ^^e2
\edef\SOuL@AtEnd{\SOuL@AtEnd\noexpand\endinput}
```

2.3 Loading packages

Package soul uses `\documentclass` to detect LaTeX.

```
\ifx\documentclass\@undefined
```

2.3.1 plain TeX

First we check, whether package soul is already loaded.

```
  \expandafter\ifx\csname SOUL@\endcsname\relax
```

In case of plain TeX package soul defines some macros in a simple manner that will break the definitions of `miniltx.tex`, for example. Therefore these macros are first saved and restored afterwards.

```
    \let\SOuL@orgDeclareRobustCommand\DeclareRobustCommand
    \let\SOuL@orgnewcommand          \newcommand
    \let\SOuL@orgDeclareOption       \DeclareOption
    \let\SOuL@orgPackageError        \PackageError
    \def\SOuL@restorelatexcmds{%
      \let\DeclareRobustCommand\SOuL@orgDeclareRobustCommand
      \let\newcommand          \SOuL@orgnewcommand
      \let\DeclareOption       \SOuL@orgDeclareOption
      \let\PackageError        \SOuL@orgPackageError
    }%
    \input soul.sty\relax
    \SOuL@restorelatexcmds
  \fi
```

`\SOUL@error` Package soul's use of `\PackageError` is replaced by `\@PackageError` of package infwarerr.

```
  \input infwarerr.sty\relax
  \let\SOuL@orgSOUL@error\SOUL@error
  \def\SOUL@error{%
    \begingroup
      \let\PackageError\@PackageError
      \SOuL@orgSOUL@error
```

```
    \endgroup
  }%
  \input etexcmds.sty\relax
```

`\@onelevel@sanitize` Define LaTeX's `\@onelevel@sanitize` if not already available.

```
  \expandafter\ifx\csname @onelevel@sanitize\endcsname\relax
    \def\@onelevel@sanitize#1{%
      \edef#1{%
        \expandafter\strip@prefix\meaning#1%
      }%
    }%
```

`\strip@prefix`

```
    \def\strip@prefix#1>{}%
  \fi
\else
```

2.3.2 LaTeX

```
  \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{soul}}%
  \ProcessOptions\relax
  \RequirePackage{soul}[2003/11/17]%
  \RequirePackage{infwarerr}[2016/05/16]%
  \RequirePackage{etexcmds}[2016/05/16]%
\fi
```

2.3.3 ε-TeX

In plain TeX command `\+` is an *outer* macro. Therefore numbers are used to avoid problems.

```
\ifetex@unexpanded
  \catcode33=14 % '!': comment
  \catcode43=9  % '+': ignore
\else
  \catcode33=9  % '!': ignore
  \catcode43=14 % '+': comment
\fi
```

2.4 Macro for redefinitions

`\SOuL@redefine`

```
\def\SOuL@redefine#1{%
  \begingroup
    \def\SOuL@cmd{#1}%
    \afterassignment\SOuL@cmdcheck
    \def\SOuL@temp
}
```

`\SOuL@cmdcheck`

```
\def\SOuL@cmdcheck{%
    \expandafter\ifx\SOuL@cmd\SOuL@temp
    \else
      \edef\SOuL@temp*{\expandafter\string\SOuL@cmd}%
      \@PackageWarningNoLine{soulutf8}{%
        Command \SOuL@temp* has changed.\MessageBreak
        Supported versions of package 'soul': 2003/11/17.\MessageBreak
```

```
        Depending on the unknown changes the redefinition\MessageBreak
        of \SOuL@temp* may not behave correctly%
      }%
    \fi
  \expandafter\endgroup
  \expandafter\def\SOuL@cmd
}
```

2.5 Redefinition of \SOUL@eval

\SOUL@eval

Macro \SOUL@eval is redefined to add detection of the first byte of a UTF-8 sequence. Because \SOUL@eval is overwritten, a warning is issued, if the contents of \SOUL@eval is not as expected.

```
\SOuL@redefine\SOUL@eval{%
```

First the expected definition.

```
  \def\SOUL@n*##1{\SOUL@scan}%
  \if\noexpand\SOUL@@\SOUL@spc
  \else
    \SOUL@ignorespacesfalse
  \fi
  \ifnum\SOUL@minus=\thr@@
    \SOUL@flushminus
  \else\ifnum\SOUL@comma=\tw@
    \SOUL@flushcomma
  \else\ifnum\SOUL@apo=\tw@
    \SOUL@flushapo
  \else\ifnum\SOUL@grave=\tw@
    \SOUL@flushgrave
  \fi\fi\fi\fi
  \ifx\SOUL@@-\else\SOUL@flushminus\fi
  \ifx\SOUL@@,\else\SOUL@flushcomma\fi
  \ifx\SOUL@@'\else\SOUL@flushapo\fi
  \ifx\SOUL@@`\else\SOUL@flushgrave\fi
  \ifx\SOUL@@-%
    \advance\SOUL@minus\@ne
  \else\ifx\SOUL@@,%
    \advance\SOUL@comma\@ne
  \else\ifx\SOUL@@'%
    \advance\SOUL@apo\@ne
  \else\ifx\SOUL@@`%
    \advance\SOUL@grave\@ne
  \else
    \SOUL@flushminus
    \SOUL@flushcomma
    \SOUL@flushapo
    \SOUL@flushgrave
    \ifx\SOUL@@\SOUL@stop
      \def\SOUL@n*{%
        \SOUL@doword
        \SOUL@eventuallyexhyphen\null
      }%
    \else\ifx\SOUL@@\par
      \def\SOUL@n*\par{\par\leavevmode\SOUL@scan}%
    \else\if\noexpand\SOUL@@\SOUL@spc
      \SOUL@doword
      \SOUL@eventuallyexhyphen\null
      \ifSOUL@ignorespaces
```

```
      \else
        \SOUL@everyspace{}%
      \fi
      \def\SOUL@n* {\SOUL@scan}%
    \else\ifx\SOUL@@\\%
      \SOUL@doword
      \SOUL@eventuallyexhyphen\null
      \SOUL@everyspace{\unskip\nobreak\hfil\break}%
      \SOUL@ignorespacestrue
    \else\ifx\SOUL@@~%
      \SOUL@doword
      \SOUL@eventuallyexhyphen\null
      \SOUL@everyspace{\nobreak}%
    \else\ifx\SOUL@@\slash
      \SOUL@doword
      \SOUL@eventuallyexhyphen{/}%
      \SOUL@exhyphen{/}%
    \else\ifx\SOUL@@\mbox
      \def\SOUL@n*{\SOUL@addprotect}%
    \else\ifx\SOUL@@\hbox
      \def\SOUL@n*{\SOUL@addprotect}%
    \else\ifx\SOUL@@\soulomit
      \def\SOUL@n*\soulomit##1{%
        \SOUL@doword
        {\spaceskip\SOUL@spaceskip##1}%
        \SOUL@scan
      }%
    \else\ifx\SOUL@@\break
      \SOUL@doword
      \break
    \else\ifx\SOUL@@\linebreak
      \SOUL@doword
      \SOUL@everyspace{\linebreak}%
    \else\ifcat\bgroup\noexpand\SOUL@@
      \def\SOUL@n*{\SOUL@addgroup{}}%
    \else\ifcat$\noexpand\SOUL@@
      \def\SOUL@n*{\SOUL@addmath}%
    \else
      \def\SOUL@n*{\SOUL@dotoken}%
    \fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi
  \fi\fi\fi\fi
  \SOUL@n*%
}{%
```

Now the redefined version follows.

```
  \def\SOUL@n*##1{\SOUL@scan}%
  \if\noexpand\SOUL@@\SOUL@spc
  \else
    \SOUL@ignorespacesfalse
  \fi
  \ifnum\SOUL@minus=\thr@@
    \SOUL@flushminus
  \else\ifnum\SOUL@comma=\tw@
    \SOUL@flushcomma
  \else\ifnum\SOUL@apo=\tw@
    \SOUL@flushapo
  \else\ifnum\SOUL@grave=\tw@
    \SOUL@flushgrave
  \fi\fi\fi\fi
```

```
\ifx\SOUL@@-\else\SOUL@flushminus\fi
\ifx\SOUL@@,\else\SOUL@flushcomma\fi
\ifx\SOUL@@'\else\SOUL@flushapo\fi
\ifx\SOUL@@`\else\SOUL@flushgrave\fi
\ifx\SOUL@@-%
  \advance\SOUL@minus\@ne
\else\ifx\SOUL@@,%
  \advance\SOUL@comma\@ne
\else\ifx\SOUL@@'%
  \advance\SOUL@apo\@ne
\else\ifx\SOUL@@`%
  \advance\SOUL@grave\@ne
\else
  \SOUL@flushminus
  \SOUL@flushcomma
  \SOUL@flushapo
  \SOUL@flushgrave
  \ifx\SOUL@@\SOUL@stop
    \def\SOUL@n*{%
        \SOUL@doword
        \SOUL@eventuallyexhyphen\null
    }%
  \else\ifx\SOUL@@\par
    \def\SOUL@n*\par{\par\leavevmode\SOUL@scan}%
  \else\if\noexpand\SOUL@@\SOUL@spc
    \SOUL@doword
    \SOUL@eventuallyexhyphen\null
    \ifSOUL@ignorespaces
    \else
      \SOUL@everyspace{}%
    \fi
    \def\SOUL@n* {\SOUL@scan}%
  \else\ifx\SOUL@@\\%
    \SOUL@doword
    \SOUL@eventuallyexhyphen\null
    \SOUL@everyspace{\unskip\nobreak\hfil\break}%
    \SOUL@ignorespacestrue
  \else\ifx\SOUL@@~%
    \SOUL@doword
    \SOUL@eventuallyexhyphen\null
    \SOUL@everyspace{\nobreak}%
  \else\ifx\SOUL@@\slash
    \SOUL@doword
    \SOUL@eventuallyexhyphen{/}%
    \SOUL@exhyphen{/}%
  \else\ifx\SOUL@@\mbox
    \def\SOUL@n*{\SOUL@addprotect}%
  \else\ifx\SOUL@@\hbox
    \def\SOUL@n*{\SOUL@addprotect}%
  \else\ifx\SOUL@@\soulomit
    \def\SOUL@n*\soulomit##1{%
      \SOUL@doword
      {\spaceskip\SOUL@spaceskip##1}%
      \SOUL@scan
    }%
  \else\ifx\SOUL@@\break
    \SOUL@doword
    \break
```

```
    \else\ifx\SOUL@@\linebreak
      \SOUL@doword
      \SOUL@everyspace{\linebreak}%
    \else\ifcat\bgroup\noexpand\SOUL@@
      \def\SOUL@n*{\SOUL@addgroup{}}%
    \else\ifcat$\noexpand\SOUL@@
      \def\SOUL@n*{\SOUL@addmath}%
    \else
```

The current token is examined to detect the start of a UTF-8 sequence.

```
      \SOuL@analyzeutfviii
      \ifcase\SOuL@octets
        \SOuL@analyzeutfviiix
      \fi
      \ifcase\SOuL@octets
        \def\SOUL@n*{\SOUL@dotoken}%
      \or % 1
      \or % 2
        \def\SOUL@n*{\SOuL@addtwooctets}%
      \or % 3
        \def\SOUL@n*{\SOuL@addthreeoctets}%
      \or % 4
        \def\SOUL@n*{\SOuL@addfouroctets}%
      \fi
    \fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi
  \fi\fi\fi\fi
  \SOUL@n*%
}
```

2.6 UTF-8 analysis

2.6.1 Help strings

```
\def\SOuL@defsanitizedstring#1#2{%
  \expandafter\def\csname SOuL@string#1\endcsname{#2}%
  \expandafter\@onelevel@sanitize\csname SOuL@string#1\endcsname
}
\SOuL@defsanitizedstring{UTFviii}{UTFviii@}
\SOuL@defsanitizedstring{octets}{@octets}
\SOuL@defsanitizedstring{two}{two}
\SOuL@defsanitizedstring{three}{three}
\SOuL@defsanitizedstring{four}{four}
\SOuL@defsanitizedstring{macrocolon}{macro:}
\SOuL@defsanitizedstring{csnameu}{csname u8-}
\SOuL@defsanitizedstring{undeferr}{utf@viii@undeferr}
\def\SOuL@stringendash{^^e2^^80^^93}
\def\SOuL@stringemdash{^^e2^^80^^94}
\def\SOuL@stringnobreakspace{^^c2^^a0}
\edef\SOuL@charhash{\string #}
\edef\SOuL@chartwo{\string 2}
\edef\SOuL@charthree{\string 3}
\def\SOuL@empty{}
```

2.6.2 Support for `utf8.def`

`\SOuL@analyzeutfviii`

```
\begingroup
  \edef\x{\endgroup
    \def\noexpand\SOuL@analyzeutfviii{%
```

```
      \noexpand\expandafter\noexpand\SOuL@checkutfviii
      \noexpand\meaning\noexpand\SOUL@@
      \SOuL@stringUTFviii\SOuL@stringoctets
      \noexpand\@nil
    }%
    \def\noexpand\SOuL@checkutfviii
      ##1\SOuL@stringUTFviii##2\SOuL@stringoctets##3\noexpand\@nil
  }%
\x{%
  \def\SOuL@temp{#2}%
  \chardef\SOuL@octets=%
      \ifx\SOuL@temp\SOuL@stringtwo
        \tw@
      \else\ifx\SOuL@temp\SOuL@stringthree
        \thr@@
      \else\ifx\SOuL@temp\SOuL@stringfour
        4 %
      \else
        \z@
      \fi\fi\fi
}
```

2.6.3 Support for `utf8x.def`

`\SOuL@analyzeutfviiix`

```
\begingroup
  \edef\x{\endgroup
    \def\noexpand\SOuL@analyzeutfviiix{%
      \noexpand\expandafter\noexpand\SOuL@checkutfviiix
      \noexpand\meaning\noexpand\SOUL@@
      \SOuL@stringmacrocolon\SOuL@charhash1{}{}{}{}%
      \SOuL@stringcsnameu\SOuL@stringundeferr
      \noexpand\@nil
    }%
```

`\SOuL@checkutfviiix`

```
    \def\noexpand\SOuL@checkutfviiix
      ##1\SOuL@stringmacrocolon\SOuL@charhash1##2##3##4##5##6%
      \SOuL@stringcsnameu##7\SOuL@stringundeferr##8\noexpand\@nil
  }%
\x{%
  \def\SOuL@temp{#7}%
  \ifx\SOuL@temp\SOuL@empty
    \chardef\SOuL@octets=\z@
  \else
    \def\SOuL@temp{#5}%
    \ifx\SOuL@temp\SOuL@charthree
      \chardef\SOuL@octets=4 %
    \else
      \def\SOuL@temp{#3}%
      \ifx\SOuL@temp\SOuL@chartwo
        \chardef\SOuL@octets=\thr@@
      \else
        \chardef\SOuL@octets=\tw@
      \fi
    \fi
  \fi
}
```

2.7 Actions for UTF-8 sequences

\SOuL@addtwooctets

```
\def\SOuL@addtwooctets#1#2{%
  \def\SOuL@temp{#1#2}%
  \@onelevel@sanitize\SOuL@temp
  \ifx\SOuL@temp\SOuL@stringnobreakspace
    \SOUL@doword
    \SOUL@eventuallyexhyphen\null
    \SOUL@everyspace{\nobreak}%
    \let\SOuL@next\SOUL@scan
  \else
    \def\SOuL@next{%
!     \SOUL@addtoken{{\noexpand#1\noexpand#2}}%
+     \SOUL@addtoken{{\etex@unexpanded{#1#2}}}%
    }%
  \fi
  \SOuL@next
}
```

\SOuL@addthreeoctets

```
\def\SOuL@addthreeoctets#1#2#3{%
  \def\SOuL@temp{#1#2#3}%
  \@onelevel@sanitize\SOuL@temp
  \ifx\SOuL@temp\SOuL@stringendash
    \SOUL@doword
    \SOUL@eventuallyexhyphen{-}%
    \SOUL@exhyphen{--}%
    \let\SOuL@next\SOUL@scan
  \else
    \ifx\SOuL@temp\SOuL@stringemdash
      \SOUL@doword
      \SOUL@eventuallyexhyphen{-}%
      \SOUL@exhyphen{---}%
      \let\SOuL@next\SOUL@scan
    \else
      \def\SOuL@next{%
!       \SOUL@addtoken{{\noexpand#1\noexpand#2\noexpand#3}}%
+       \SOUL@addtoken{{\etex@unexpanded{#1#2#3}}}%
      }%
    \fi
  \fi
  \SOuL@next
}
```

\SOuL@addfouroctets

```
\def\SOuL@addfouroctets#1#2#3#4{%
! \SOUL@addtoken{{\noexpand#1\noexpand#2\noexpand#3\noexpand#4}}%
+ \SOUL@addtoken{{\etex@unexpanded{#1#2#3#4}}}%
}
```

2.7.1 Redefinition of \SOUL@splittoken

\SOUL@splittoken

Macro \SOUL@splittoken separates the first token or token group from a word and redefines the word to contain the remaining tokens. However if the remaining tokens are a token group, then the curly braces will be removed and the token group is splitted by the next call of \SOUL@splittoken. The redefinition avoids the removal of curly braces around the remaining tokens.

```
\SOuL@redefine\SOUL@splittoken#1#2\SOUL@stop{%
  \global\SOUL@token={#1}%
  \global\SOUL@word={#2}%
}#1{%
  \global\SOUL@token={#1}%
  \SOuL@remainingtoken\relax
}
```

\SOuL@remainingtoken

```
\def\SOuL@remainingtoken#1\SOUL@stop{%
  \global\SOUL@word=\expandafter{\@gobble#1}%
}
```

2.8 Patches

The fixed \SOUL@splittoken allows to remove the double sets of curly braces in other macros of package soul. The benefit is that implicite kernings are more often detected and fixes a bug in package soul. The disadvantage is incompatibility. The width of the resulting strings may change.

\SOUL@flushcomma

```
\SOuL@redefine\SOUL@flushcomma{%
  \ifcase\SOUL@comma
  \or
    \edef\x{\SOUL@word={\the\SOUL@word,}}\x
  \or
    \edef\x{\SOUL@word={\the\SOUL@word{{,,}}}}\x
  \fi
  \SOUL@comma\z@
}{%
  \ifcase\SOUL@comma
  \or
    \edef\x{\SOUL@word={\the\SOUL@word,}}\x
  \or
    \edef\x{\SOUL@word={\the\SOUL@word{,,}}}\x
  \fi
  \SOUL@comma\z@
}
```

\SOUL@flushapo

```
\SOuL@redefine\SOUL@flushapo{%
  \ifcase\SOUL@apo
  \or
    \edef\x{\SOUL@word={\the\SOUL@word'}}\x
  \or
    \edef\x{\SOUL@word={\the\SOUL@word{{''}}}}\x
  \fi
  \SOUL@apo\z@
}{%
  \ifcase\SOUL@apo
  \or
    \edef\x{\SOUL@word={\the\SOUL@word'}}\x
  \or
    \edef\x{\SOUL@word={\the\SOUL@word{''}}}\x
  \fi
  \SOUL@apo\z@
}
```

\SOUL@flushgrave

```
\SOuL@redefine\SOUL@flushgrave{%
  \ifcase\SOUL@grave
  \or
    \edef\x{\SOUL@word={\the\SOUL@word‘}}\x
  \or
    \edef\x{\SOUL@word={\the\SOUL@word{{‘‘}}}}\x
  \fi
  \SOUL@grave\z@
}{%
  \ifcase\SOUL@grave
  \or
    \edef\x{\SOUL@word={\the\SOUL@word‘}}\x
  \or
    \edef\x{\SOUL@word={\the\SOUL@word{‘‘}}}\x
  \fi
  \SOUL@grave\z@
}
```

\SOUL@addgroup

```
\SOuL@redefine\SOUL@addgroup#1#2{%
  {%
    \let\protect\noexpand
    \edef\x{%
      \global\SOUL@word={%
        \the\SOUL@word
        {{\noexpand#1#2}}%
      }%
    }%
    \x
  }%
  \SOUL@scan
}#1#2{%
  \begingroup
    \let\protect\noexpand
    \edef\x{\endgroup
      \SOUL@word={%
        \the\SOUL@word
!       {\noexpand#1{#2}}%
+       {\etex@unexpanded{#1{#2}}}%
      }%
    }%
  \x
  \SOUL@scan
}
```

\SOUL@addmath

```
\SOuL@redefine\SOUL@addmath$#1$%{
  {%
    \let\protect\noexpand
    \edef\x{%
      \global\SOUL@word={%
        \the\SOUL@word
        {{\hbox{$#1$}}}%
      }%
    }%
    \x
  }%
```

```
  \SOUL@scan
}$#1${%
  \begingroup
    \let\protect\noexpand
    \edef\x{\endgroup
      \SOUL@word={%
        \the\SOUL@word
!       {\hbox{$#1$}}%
+       {\etex@unexpanded{\hbox{$#1$}}}%
      }%
    }%
  \x
  \SOUL@scan
}
```

\SOUL@addprotect

```
\SOuL@redefine\SOUL@addprotect#1#2{%
  {%
    \let\protect\noexpand
    \edef\x{%
      \global\SOUL@word={%
        \the\SOUL@word
        {{\hbox{#2}}}%
      }%
    }%
    \x
  }%
  \SOUL@scan
}#1#2{%
  \begingroup
    \let\protect\noexpand
    \edef\x{\endgroup
      \SOUL@word={%
        \the\SOUL@word
!       {\hbox{#2}}%
+       {\etex@unexpanded{\hbox{#2}}}%
      }%
    }%
  \x
  \SOUL@scan
}
```

\SOUL@addtoken

```
+ \SOuL@redefine\SOUL@addtoken#1{%
+   \edef\x{%
+     \SOUL@word={%
+       \the\SOUL@word
+       \noexpand#1%
+     }%
+   }%
+   \x
+   \SOUL@scan
+ }#1{%
+   \edef\x{%
+     \SOUL@word={%
+       \the\SOUL@word
+       \etex@unexpanded{#1}%
+     }%
```

```
+    }%
+    \x
+    \SOUL@scan
+  }%

\SOuL@AtEnd%
```

⟨/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
}
\def\RangeCatcodeCheck#1#2#3{%
  \count@=#1\relax
  \loop
    \ifnum#3=\catcode\count@
    \else
      \errmessage{%
        Character \the\count@\space
        with wrong catcode \the\catcode\count@\space
        instead of \number#3%
      }%
    \fi
  \ifnum\count@<#2\relax
    \advance\count@ 1 %
  \repeat
}
\def\space{ }
\expandafter\ifx\csname LoadCommand\endcsname\relax
  \def\LoadCommand{\input soulutf8.sty\relax}%
\fi
\def\Test{%
  \RangeCatcodeInvalid{0}{47}%
  \RangeCatcodeInvalid{58}{64}%
  \RangeCatcodeInvalid{91}{96}%
  \RangeCatcodeInvalid{123}{255}%
  \catcode`\@=12 %
  \catcode`\\=0 %
  \catcode`\%=14 %
  \LoadCommand
  \RangeCatcodeCheck{0}{36}{15}%
  \RangeCatcodeCheck{37}{37}{14}%
  \RangeCatcodeCheck{38}{47}{15}%
  \RangeCatcodeCheck{48}{57}{12}%
  \RangeCatcodeCheck{58}{63}{15}%
  \RangeCatcodeCheck{64}{64}{12}%
  \RangeCatcodeCheck{65}{90}{11}%
  \RangeCatcodeCheck{91}{91}{15}%
  \RangeCatcodeCheck{92}{92}{0}%
  \RangeCatcodeCheck{93}{96}{15}%
  \RangeCatcodeCheck{97}{122}{11}%
  \RangeCatcodeCheck{123}{255}{15}%
  \RestoreCatcodes
}
\Test
\csname @@end\endcsname
\end
```

⟨/test1⟩

3.2 LaTeX tests

⟨*test2 | test3 | test4 | test5⟩

```
⟨noetex⟩\let\SavedUnexpanded\unexpanded
⟨noetex⟩\let\unexpanded\UNDEFINED
\NeedsTeXFormat{LaTeX2e}
\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage{lmodern}
\usepackage{textcomp}
\usepackage[ngerman]{babel}
\expandafter\def\csname ver@hypdestopt.sty\endcsname{}
\usepackage[unicode]{hyperref}
\usepackage{bookmark}
\usepackage{color}
\usepackage{soul}[2003/11/17]
\makeatletter
\newcommand*{\@ifDefined}[1]{%
  \@ifundefined{#1}{%
    \@latex@error{%
      \expandafter\noexpand\csname#1\endcsname undefined%
    }\@ehc
  }%
}
\newcommand*{\SaveOrg}[1]{%
  \@ifdefinable{ORG@SOUL@#1}{%
    \@ifDefined{SOUL@#1}{%
      \expandafter\let\csname ORG@SOUL@#1\expandafter\endcsname
                      \csname SOUL@#1\endcsname
    }%
  }%
}
\newcommand*{\RestoreOrg}[1]{%
  \@ifDefined{ORG@SOUL@#1}{%
    \@ifDefined{SOUL@#1}{%
      \expandafter\let\csname SOUL@#1\expandafter\endcsname
                      \csname ORG@SOUL@#1\endcsname
    }%
  }%
}
\makeatother
\SaveOrg{eval}
\SaveOrg{splittoken}
\SaveOrg{flushcomma}
\SaveOrg{flushapo}
\SaveOrg{grave}
\SaveOrg{addgroup}
\SaveOrg{addmath}
\SaveOrg{addprotect}
\SaveOrg{addtoken}
\usepackage{soulutf8}[2016/05/16]

⟨noetex⟩\let\unexpanded\SavedUnexpanded
```

Input encoding `utf8`.

```
⟨*utf8⟩
\usepackage[utf8]{inputenc}
\DeclareUnicodeCharacter{00A0}{\nobreakspace}
⟨/utf8⟩
```

Input encoding `utf8x`. Package `ucs` has a nasty bug, that it redefines `\show`.

```
⟨*utf8x⟩
\let\origshow\show
```

```
\usepackage[utf8x]{inputenc}
\AtBeginDocument{%
  \let\show\origshow
}
```
⟨/utf8x⟩

3.2.1 Driver definitions

```
\makeatletter
```

\an The definition for \an is taken from the documentation of package soul.

```
\DeclareRobustCommand*{\an}{%
  \def\SOUL@preamble{$^{^P}$}%
  \def\SOUL@everyspace##1{##1\texttt{\char`\ }\linebreak[2]}%
  \def\SOUL@postamble{$^{^E}$}%
  \def\SOUL@everyhyphen{$^{^-}$\linebreak[1]}%
  \def\SOUL@everyexhyphen##1{##1$^{^=}$}%
  \def\SOUL@everysyllable{$^{^S}$}%
  \def\SOUL@everytoken{\the\SOUL@token$^{^T}$}%
  \def\SOUL@everylowerthan{$^{^L}$}%
  \SOUL@
}
```

\sy The definition for \sy is taken from the documentation of package soul.

```
\DeclareRobustCommand*{\sy}{%
  \SOUL@setup
  \def\SOUL@preamble{%
    \lefthyphenmin=\z@
    \righthyphenmin=\z@
  }%
  \def\SOUL@everyspace##1{##1\space}%
  \def\SOUL@everysyllable{\the\SOUL@syllable}%
  \def\SOUL@everyhyphen{%
    \discretionary{%
      \SOUL@setkern\SOUL@hyphkern
      \SOUL@sethyphenchar
    }{}{%
      \hbox{\kern1pt$\cdot$}%
    }%
  }%
  \def\SOUL@everyexhyphen##1{%
    \SOUL@setkern\SOUL@hyphkern
    \hbox{##1}%
    \discretionary{}{}{%
      \SOUL@setkern\SOUL@charkern
    }%
  }%
  \SOUL@
}
```

\soulH

```
\DeclareRobustCommand*{\soulH}{%
  \def\SOUL@preamble{%
    \gdef\SOULH{}%
    \soulHadd{[}%
  }%
  \def\SOUL@everyspace##1{%
    \soulHadd{Sp}%
  }%
```

```
  \def\SOUL@postamble{%
    \soulHadd{]}%
  }%
  \def\SOUL@everyhyphen{%
    \soulHadd{Hy}%
  }%
  \def\SOUL@everyexhyphen##1{%
    \soulHadd{Exhy}%
  }%
  \def\SOUL@everysyllable{%
    \soulHadd{Sy}%
  }%
  \def\SOUL@everytoken{%
    \soulHadd{T}%
  }%
  \def\SOUL@everylowerthan{%
    \soulHadd{<}%
  }%
  \SOUL@
}
\newcommand*{\soulHadd}[1]{%
  #1%
  \expandafter\gdef\expandafter\soulHstring\expandafter{%
    \soulHstring
    #1%
  }%
}
```

3.2.2 Test macros

```
\usepackage{qstest}
\IncludeTests{*}
\LogTests{log}{*}{*}

\newsavebox\boxA
\newsavebox\boxB
\newsavebox\boxC
```

\TestDriver

```
\newcommand*{\TestDriver}[4]{%
  \subsection{Driver #1}%
  \@PackageInfo{soulutf8-test}{* Driver #1}%
  \expandafter\let\expandafter\testdriver\csname #1\endcsname
  \def\teststringA{#3}%
  \def\teststringB{#4}%
  \begin{qstest}{string-#2-driver-#1}{string,#2,driver,#1}%
    \@PackageInfo{soulutf8-test}{* Version: soulutf8}%
    \global\let\soulHstring\@empty
    \setbox0=\hbox{\expandafter\testdriver\expandafter{\teststringA}}%
    \let\HstringA\soulHstring
    \SetTestVBox{\boxA}{\teststringA}%
    %
    \@PackageInfo{soulutf8-test}{* Version: soul + patch}%
    \global\let\soulHstring\@empty
    \RestoreOrg{eval}%
    \setbox2=\hbox{\expandafter\testdriver\expandafter{\teststringB}}%
    \let\HstringB\soulHstring
    \SetTestVBox{\boxB}{\teststringB}%
    %
```

```
    \@PackageInfo{soulutf8-test}{* Version: original soul}%
    \global\let\soulHstring\@empty
    \RestoreOrg{splittoken}%
    \RestoreOrg{flushcomma}%
    \RestoreOrg{flushapo}%
    \RestoreOrg{grave}%
    \RestoreOrg{addgroup}%
    \RestoreOrg{addmath}%
    \RestoreOrg{addprotect}%
    \RestoreOrg{addtoken}%
    \setbox4=\hbox{\expandafter\testdriver\expandafter{\teststringB}}%
    \let\HstringC\soulHstring
    \SetTestVBox{\boxC}{\teststringB}%
    %
    \TestInfo{0}{soulutf8}%
    \TestInfo{2}{soul + patch}%
    \TestInfo{4}{original soul}%
    \TestInfoVBox
    \showboxdepth=\maxdimen
    \showboxbreadth=\maxdimen
    \tracingonline=\@ne
    \Expect*{\HstringA}*{\HstringB}%
    \Expect*{\HstringA}*{\HstringC}%
    \Expect*{\the\wd0}*{\the\wd2}%
    \ifdim\wd0=\wd2 %
    \else
      \showbox0 %
      \showbox2 %
    \fi
    \Expect*{\the\ht0}*{\the\ht2}%
    \Expect*{\the\dp0}*{\the\dp2}%
    \Expect*{\the\wd\boxB}*{\the\wd\boxA}%
    \Expect*{\the\wd\boxC}*{\the\wd\boxA}%
    \Expect*{\the\ht\boxB}*{\the\ht\boxA}%
    \ifdim\ht\boxB=\ht\boxA
    \else
      \showbox\boxA
      \showbox\boxB
    \fi
    \Expect*{\the\ht\boxC}*{\the\ht\boxA}%
    \Expect*{\the\dp\boxB}*{\the\dp\boxA}%
    \Expect*{\the\dp\boxC}*{\the\dp\boxA}%
  \end{qstest}%
}
```

\SetTestVBox

```
\newcommand*{\SetTestVBox}[2]{%
  \setbox#1=\vbox{%
    \hsize=0pt\relax
    \hfuzz=100mm\relax
    \expandafter\testdriver\expandafter{#2}%
  }%
}
```

\TestInfo

```
\newcommand*{\TestInfo}[2]{%
  \par
  \noindent
```

```
  \copy#1\relax
  \hfill
  \begingroup
    \small
    \ifdim\wd0=\wd2\relax
      \ifdim\wd0=\wd4\relax
      \else
        \ifnum#1=2 %
          \raisebox{.5\baselineskip}[0pt][0pt]{%
            \llap{$\}$ }%
            \rlap{width = \the\wd#1\relax\space$\{$}%
            \phantom{width = 999.99999pt}%
            \qquad
          }%
        \else
          \ifnum#1=4 %
            \rlap{width = \the\wd#1}%
            \phantom{width = 999.99999pt}%
            \qquad
          \fi
        \fi
      \fi
    \else
      \begingroup
        \bfseries
        \rlap{width = \the\wd#1}%
        \phantom{width = 999.99999pt}%
      \endgroup
      \qquad
    \fi
    \rlap{(#2)}%
    \phantom{(original soul)}%
  \endgroup
  \par
}
```

\TestInfoVBox

```
\newcommand*{\TestInfoVBox}{%
  \par
  \noindent
  \mbox{\vrule\relax\space\copy\boxA}%
  \ifdim\ht\boxA=\ht\boxB
  \else
    \hfill
    \begingroup
      \small
      \rlap{(soulutf8)}%
      \phantom{(soulutf8)}%
    \endgroup
    \par
    \noindent
    \mbox{\vrule\relax\space\copy\boxB}%
    \hfill
    \begingroup
      \small
      \rlap{(soul)}%
      \phantom{(soulutf8)}%
    \endgroup
```

```
    \fi
    \par
}
```

\Test

```
\makeatletter
\newcommand*{\Test}[3]{%
  \clearpage
  \begingroup
    \def\linebreak{\textbackslash linebreak }%
    \def\\{\textbackslash\textbackslash}%
    \def\soulomit##1{%
      \textbackslash soulomit\textbraceleft##1\textbraceright
    }%
    \def\mbox##1{%
      \textbackslash mbox\textbraceleft##1\textbraceright
    }%
    \def\abc{\textbackslash abc }%
    \def\texttt##1{%
      \textbackslash texttt\textbraceleft##1\textbraceright
    }%
    \def\c##1{%
      \textbackslash c\textbraceleft##1\textbraceright
    }%
⟨*utf8x⟩
    \count@=128 %
    \loop
    \ifnum\count@<256 %
      \lccode`\~=\count@
      \lowercase{%
        \def~{\noexpand~}%
      }%
      \advance\count@\@ne
    \repeat
⟨/utf8x⟩
    \csname protected@edef\endcsname\x{\endgroup
      \noexpand\section{String #1: #2}%
    }%
  \x
  \TestDriver{so}{#1}{#2}{#3}%
  \TestDriver{caps}{#1}{#2}{#3}%
  \TestDriver{ul}{#1}{#2}{#3}%
  \TestDriver{st}{#1}{#2}{#3}%
  \TestDriver{hl}{#1}{#2}{#3}%
  \TestDriver{an}{#1}{#2}{#3}%
  \TestDriver{sy}{#1}{#2}{#3}%
  \TestDriver{soulH}{#1}{#2}{#3}%
}
\makeatother
```

3.2.3 Testing

```
\begin{document}
  \Test A{}{}%
  \Test B{Hello World}{Hello World}%
  \Test C{%
    Umlauts: ^^c3^^84^^c3^^96^^c3^^9c^^c3^^a4^^c3^^b6^^c3^^bc^^c3^^9f%
  }{%
```

```
    Umlauts: \"A\"O\"U\"a\"o\"u\ss
  }%
  \Test D{Euro: ^^e2^^82^^ac}{Euro: \texteuro}%
  \Test E{Stra^^c3^^9fe}{Stra\ss e}%
  \Test F{R^^c3^^b6mer}{R\"omer}%
  \Test G{A^^e2^^80^^93B^^e2^^80^^94C}{A--B---C}%
  \Test H{X^^c2^^a0Y}{X~Y}%
  \Test I{A^^c5^^a4}{A\v T}%
  \Test J{A,B,,C,,,D,,,,E}{A,B,,C,,,D,,,,E}%
  \Test K{A'B''C'''D''''E}{A'B''C'''D''''E}%
  \Test L{A`B``C```D````E}{A`B``C```D````E}%
  \Test M{A$x$B}{A$x$B}%
  \Test N{A\mbox{abc}B}{A\mbox{abc}B}%
  \Test O{,,TA''}{,,TA''}%
  \Test P{{!`}Hola{?`}}{{!`}Hola{?`}}%
  \Test Q{re-sent\slash x}{re-sent\slash x}%
  \Test R{new\\line}{new\\line}%
  \Test S{foo\linebreak bar}{foo\linebreak bar}%
  \Test T{Virchow{sche}}{Virchow{sche}}%
  \Test U{A\soulomit{foo}B}{A\soulomit{foo}B}%
  \Test V{A\texttt{xyz}B}{A\texttt{xyz}B}%
  \Test W{ffi}{ffi}%
  \Test X{f{}f{}i}{f{}f{}i}%
  \Test Y{f\mbox{}f\mbox{}i}{f\mbox{}f\mbox{}i}%
  \Test Z{\c{\"a}}{{\c{\"a}}}%
⟨*test2 | test3⟩
  \let\abcx\UNDEFINED
  \def\abc{%
    \begingroup
      \def\abcx{x}%
      \abcx
    \endgroup
  }%
  \Test{AA}{[{x\abc x}]}{[{xxx}]}%
⟨/test2 | test3⟩
\end{document}
⟨/test2 | test3 | test4 | test5⟩
```

4 Installation

4.1 Download

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

CTAN:macros/latex/contrib/oberdiek/soulutf8.dtx The source file.

CTAN:macros/latex/contrib/oberdiek/soulutf8.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] http://ctan.org/pkg/soulutf8

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 soulutf8.dtx
```

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

```
soulutf8.sty            → tex/generic/oberdiek/soulutf8.sty
soulutf8.pdf            → doc/latex/oberdiek/soulutf8.pdf
test/soulutf8-test1.tex → doc/latex/oberdiek/test/soulutf8-test1.tex
test/soulutf8-test2.tex → doc/latex/oberdiek/test/soulutf8-test2.tex
test/soulutf8-test3.tex → doc/latex/oberdiek/test/soulutf8-test3.tex
test/soulutf8-test4.tex → doc/latex/oberdiek/test/soulutf8-test4.tex
test/soulutf8-test5.tex → doc/latex/oberdiek/test/soulutf8-test5.tex
soulutf8.dtx            → source/latex/oberdiek/soulutf8.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 soulutf8.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{soulutf8.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 soulutf8.dtx
makeindex -s gind.ist soulutf8.idx
pdflatex soulutf8.dtx
makeindex -s gind.ist soulutf8.idx
pdflatex soulutf8.dtx
```

5 Catalogue

The following XML file can be used as source for the TeX Catalogue. The elements `caption` and `description` are imported from the original XML file from the Catalogue. The name of the XML file in the Catalogue is `soulutf8.xml`.

```
⟨*catalogue⟩
<?xml version='1.0' encoding='us-ascii'?>
<!DOCTYPE entry SYSTEM 'catalogue.dtd'>
<entry datestamp='$Date$' modifier='$Author$' id='soulutf8'>
  <name>soulutf8</name>
  <caption>Permit use of UTF-8 characters in soul.</caption>
  <authorref id='auth:oberdiek'/>
  <copyright owner='Heiko Oberdiek' year='2007'/>
  <license type='lppl1.3'/>
  <version number='1.1'/>
  <description>
    This package extends package <xref refid='soul'>soul</xref> and adds
    some support for UTF-8. Namely the input encodings in
    &#x2018;<tt>utf8.def</tt>&#x2019; from package
    <xref refid='inputenc'>inputenc</xref> and
    &#x2018;<tt>utf8x.def</tt>&#x2019; from package
    <xref refid='unicode'>ucs</xref> are supported.
    <p/>
    The package is part of the <xref refid='oberdiek'>oberdiek</xref>
    bundle.
  </description>
  <documentation details='Package documentation'
      href='ctan:/macros/latex/contrib/oberdiek/soulutf8.pdf'/>
  <ctan file='true' path='/macros/latex/contrib/oberdiek/soulutf8.dtx'/>
  <miktex location='oberdiek'/>
  <texlive location='oberdiek'/>
  <install path='/macros/latex/contrib/oberdiek/oberdiek.tds.zip'/>
</entry>
⟨/catalogue⟩
```

6 References

[1] Melchior Franz: *The soul package*; 2003/11/17;
CTAN:macros/latex/contrib/soul/.

[2] Dominique P. G. Unruh: *ucs.sty – Unicode Support*; 2004/10/17;
CTAN:macros/latex/contrib/unicode/.

[3] Frank Mittelbach, Chris Rowley: *Providing some UTF-8 support via inputenc*; 2006/03/30;
CTAN:macros/latex/base/utf8ienc.dtx.

7 History

[2007/09/09 v1.0]

- First version.

[2016/05/16 v1.1]

- Documentation updates.

8 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.

Symbols

\" 1074, 1078, 1098
\# 656
\% 732
\@ 657, 730
\@PackageError 148
\@PackageInfo 893, 898, 904, 911
\@PackageWarningNoLine 185
\@ehc 771
\@empty 899, 905, 912
\@firstofone 665, 668
\@gobble 504, 662, 670
\@ifDefined 767, 776, 783, 784
\@ifdefinable 775
\@ifundefined 768
\@latex@error 769
\@ne 215, 217, 219, 221, 300, 302, 304, 306, 931, 1051
\@nil 404, 407, 429, 433
\@onelevel@sanitize 153, 381, 455, 471
\@undefined 58, 129
\\ 242, 327, 731, 1029, 1090
\{ 654, 975
\} 655, 974
\~ 1047
\␣ 816

A

\abc 1036, 1101, 1107
\abcx 1100, 1103, 1104
\active 121
\advance . . 215, 217, 219, 221, 300, 302, 304, 306, 695, 703, 718, 1051
\afterassignment 178
\aftergroup 29
\an 814
\AtBeginDocument 809

B

\baselineskip 973
\begin 897, 1068
\bfseries 989
\body 674, 678
\boxA 888, 902, 942, 943, 944, 945, 947, 950, 951, 952, 1003, 1004
\boxB 889, 909, 942, 944, 945, 948, 951, 1004, 1014
\boxC 890, 923, 943, 950, 952
\break 245, 265, 267, 330, 350, 352

C

\c 1040, 1098
\catcode 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 33, 34, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45,

46, 47, 48, 49, 69, 70, 72, 73, 74, 78, 79, 80, 81, 82, 83, 84, 87, 88, 90, 91, 92, 93, 97, 99, 169, 170, 172, 173, 654, 655, 656, 657, 692, 701, 709, 713, 730, 731, 732
\cdot 838
\char 816
\chardef 411, 438, 442, 446, 448
\clearpage 1026
\copy 965, 1003, 1014
\count@ 659, 688, 692, 694, 695, 699, 701, 702, 703, 707, 709, 712, 713, 717, 718, 1044, 1046, 1047, 1051
\countdef 659
\csname 14, 21, 50, 66, 76, 130, 153, 380, 381, 658, 661, 664, 667, 722, 749, 761, 770, 777, 778, 785, 786, 894, 1054
\CurrentOption 162

D

\DeclareOption 133, 138, 162
\DeclareRobustCommand 131, 136, 814, 825, 850
\DeclareUnicodeCharacter 804
\discretionary 834, 844
\documentclass 129, 756
\dp 941, 951, 952

E

\empty 17, 18
\end 750, 953, 1109
\endcsname 14, 21, 50, 66, 76, 130, 153, 380, 381, 658, 661, 664, 667, 722, 749, 761, 770, 777, 778, 785, 786, 894, 1054
\endinput 29, 128
\endlinechar 4, 35, 71, 77, 89
\errmessage 711
\etex@unexpanded 464, 486, 494, 576, 601, 626, 645
\Expect 932, 933, 934, 940, 941, 942, 943, 944, 950, 951, 952

G

\gdef 852, 880

H

\hbox . . 257, 342, 588, 600, 601, 613, 625, 626, 838, 843, 900, 907, 921
\hfil 245, 330
\hfill 966, 1006, 1015
\hfuzz 958
\hsize 957
\HstringA 901, 932, 933
\HstringB 908, 932
\HstringC 922, 933
\ht 940, 944, 945, 950, 1004

I

\if 197, 234, 282, 319
\ifcase 362, 365, 507, 515, 524, 532, 541, 549
\ifcat 271, 273, 356, 358
\ifdim 935, 945, 969, 970, 1004
\ifetex@unexpanded 168
\ifnum 201, 203, 205, 207, 286, 288, 290, 292, 694, 702, 709, 717, 972, 980, 1046
\ifSOUL@ignorespaces 237, 322
\ifx 15, 18, 21, 50, 58, 61, 129, 130, 153, 182, 210, 211, 212, 213, 214, 216, 218, 220, 227, 232, 242, 247, 251, 255, 257, 259, 265, 268, 295, 296, 297, 298, 299, 301, 303, 305, 312, 317, 327, 332, 336, 340, 342, 344, 350, 353, 412, 414, 416, 437, 441, 445, 456, 472, 478, 658, 661, 664, 667, 722
\immediate 23, 52
\IncludeTests 886
\input 141, 144, 152, 723
\iterate 675, 677, 679

K

\kern 838

L

\lccode 1047
\leavevmode 233, 318
\lefthyphenmin 828
\linebreak 268, 270, 353, 355, 816, 818, 1028, 1091
\llap 974
\LoadCommand 723, 733
\LogTests 887
\loop 673, 689, 700, 708, 1045
\lowercase 1048

M

\makeatletter 766, 813, 1024
\makeatother 790, 1067
\maxdimen 929, 930
\mbox 255, 340, 1003, 1014, 1033, 1086, 1097
\meaning 156, 402, 426
\MessageBreak 186, 187, 188

N

\NeedsTeXFormat 755
\newcommand 132, 137, 767, 774, 782, 878, 891, 955, 962, 1000, 1025
\newsavebox 888, 889, 890
\next 679, 681, 683
\nobreak 245, 250, 330, 335, 459
\nobreakspace 804
\noindent 964, 1002, 1013

\null 230, 236, 244, 249, 315, 321, 329, 334, 458
\number 714

O

\origshow 807, 810

P

\PackageError 134, 139, 148
\PackageInfo 26
\par 232, 233, 317, 318, 963, 998, 1001, 1012, 1022
\PassOptionsToPackage 162
\phantom 976, 982, 991, 996, 1010, 1019
\ProcessOptions 163
\protect . . 559, 571, 584, 596, 609, 621
\ProvidesPackage 19, 67

Q

\qquad 977, 983, 993

R

\raisebox 973
\RangeCatcodeCheck 706, 734, 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745
\RangeCatcodeInvalid 698, 726, 727, 728, 729
\repeat . . 673, 685, 696, 704, 719, 1052
\RequirePackage 164, 165, 166
\RestoreCatcodes . . 687, 690, 691, 746
\RestoreOrg 782, 906, 913, 914, 915, 916, 917, 918, 919, 920
\righthyphenmin 829
\rlap . . . 975, 981, 990, 995, 1009, 1018

S

\SavedUnexpanded 753, 801
\SaveOrg 774, 791, 792, 793, 794, 795, 796, 797, 798, 799
\section 1055
\setbox 900, 907, 921, 956
\SetTestVBox 902, 909, 923, 955
\show 807, 810
\showbox 937, 938, 947, 948
\showboxbreadth 930
\showboxdepth 929
\slash 251, 336, 1089
\small 968, 1008, 1017
\SOUL@ 823, 848, 876
\SOUL@@ 197, 210, 211, 212, 213, 214, 216, 218, 220, 227, 232, 234, 242, 247, 251, 255, 257, 259, 265, 268, 271, 273, 282, 295, 296, 297, 298, 299, 301, 303, 305, 312, 317, 319, 327, 332, 336, 340, 342, 344, 350, 353, 356, 358, 402, 426
\SOuL@addfouroctets 373, 492
\SOUL@addgroup 272, 357, 557
\SOUL@addmath 274, 359, 582
\SOUL@addprotect 256, 258, 341, 343, 607
\SOuL@addthreeoctets 371, 469
\SOUL@addtoken 463, 464, 485, 486, 493, 494, 632
\SOuL@addtwooctets 369, 453
\SOuL@analyzeutfviii 361, 398
\SOuL@analyzeutfviiix 363, 422
\SOUL@apo 205, 219, 290, 304, 524, 530, 532, 538
\SOuL@AtEnd 95, 96, 128, 651
\SOuL@charhash 394, 427, 432
\SOUL@charkern 845
\SOuL@charthree 396, 441
\SOuL@chartwo 395, 445
\SOuL@checkutfviii 401, 406
\SOuL@checkutfviiix 425, 431
\SOuL@cmd 177, 182, 184, 193
\SOuL@cmdcheck 178, 181
\SOUL@comma 203, 217, 288, 302, 507, 513, 515, 521
\SOuL@defsanitizedstring 379, 383, 384, 385, 386, 387, 388, 389, 390
\SOUL@dotoken 276, 366
\SOUL@doword 229, 235, 243, 248, 252, 261, 266, 269, 314, 320, 328, 333, 337, 346, 351, 354, 457, 473, 479
\SOuL@empty 397, 437
\SOUL@error 144
\SOUL@eval 195
\SOUL@eventuallyexhyphen 230, 236, 244, 249, 253, 315, 321, 329, 334, 338, 458, 474, 480
\SOUL@everyexhyphen . . . 819, 841, 864
\SOUL@everyhyphen 818, 833, 861
\SOUL@everylowerthan 822, 873
\SOUL@everyspace 239, 245, 250, 270, 324, 330, 335, 355, 459, 816, 831, 855
\SOUL@everysyllable . . . 820, 832, 867
\SOUL@everytoken 821, 870
\SOUL@exhyphen 254, 339, 475, 481
\SOUL@flushapo 206, 212, 225, 291, 297, 310, 523
\SOUL@flushcomma 204, 211, 224, 289, 296, 309, 506
\SOUL@flushgrave 208, 213, 226, 293, 298, 311, 540
\SOUL@flushminus 202, 210, 223, 287, 295, 308
\SOUL@grave 207, 221, 292, 306, 541, 547, 549, 555
\SOUL@hyphkern 835, 842
\SOUL@ignorespacesfalse . . . 199, 284
\SOUL@ignorespacestrue 246, 331
\SOUL@minus 201, 215, 286, 300

\SOUL@n 196, 228, 233, 241, 256, 258, 260, 272, 274, 276, 279, 281, 313, 318, 326, 341, 343, 345, 357, 359, 366, 369, 371, 373, 377
\SOuL@next 460, 462, 467, 476, 482, 484, 490
\SOuL@octets 362, 365, 411, 438, 442, 446, 448
\SOuL@orgDeclareOption 133, 138
\SOuL@orgDeclareRobustCommand 131, 136
\SOuL@orgnewcommand 132, 137
\SOuL@orgPackageError 134, 139
\SOuL@orgSOUL@error 145, 149
\SOUL@postamble 817, 858
\SOUL@preamble 815, 827, 851
\SOuL@redefine 175, 195, 496, 506, 523, 540, 557, 582, 607, 632
\SOuL@remainingtoken 501, 503
\SOuL@restorelatexcmds 135, 142
\SOUL@scan 196, 233, 241, 263, 281, 318, 326, 348, 460, 476, 482, 568, 580, 593, 605, 618, 630, 640, 649
\SOUL@sethyphenchar 836
\SOUL@setkern 835, 842, 845
\SOUL@setup 826
\SOUL@spaceskip 262, 347
\SOUL@spc 197, 234, 282, 319
\SOUL@splittoken 496
\SOUL@stop 227, 312, 496, 503
\SOuL@stringcsnameu 428, 433
\SOuL@stringemdash 392, 478
\SOuL@stringendash 391, 472
\SOuL@stringfour 416
\SOuL@stringmacrocolon 427, 432
\SOuL@stringnobreakspace . . . 393, 456
\SOuL@stringoctets 403, 407
\SOuL@stringthree 414
\SOuL@stringtwo 412
\SOuL@stringundeferr 428, 433
\SOuL@stringUTFviii 403, 407
\SOUL@syllable 832
\SOuL@temp 179, 182, 184, 186, 189, 410, 412, 414, 416, 436, 437, 440, 441, 444, 445, 454, 455, 456, 470, 471, 472, 478
\SOUL@token 497, 500, 821
\SOUL@word 498, 504, 509, 511, 517, 519, 526, 528, 534, 536, 543, 545, 551, 553, 561, 562, 573, 574, 586, 587, 598, 599, 611, 612, 623, 624, 634, 635, 643, 644
\SOULH 852
\soulH 850
\soulHadd 853, 856, 859, 862, 865, 868, 871, 874, 878
\soulHstring 880, 881, 899, 901, 905, 908, 912, 922
\soulomit 259, 260, 344, 345, 1030, 1093
\space 712, 713, 721, 831, 975, 1003, 1014
\spaceskip 262, 347
\ss 1074, 1077
\strip@prefix 156, 159
\subsection 892
\sy 825

T

\Test 725, 748, 1024, 1069, 1070, 1071, 1076, 1077, 1078, 1079, 1080, 1081, 1082, 1083, 1084, 1085, 1086, 1087, 1088, 1089, 1090, 1091, 1092, 1093, 1094, 1095, 1096, 1097, 1098, 1107
\TestDriver 891, 1058, 1059, 1060, 1061, 1062, 1063, 1064, 1065
\testdriver . . . 894, 900, 907, 921, 959
\TestInfo 925, 926, 927, 962
\TestInfoVBox 928, 1000
\teststringA 895, 900, 902
\teststringB . . 896, 907, 909, 921, 923
\textbackslash 1028, 1029, 1031, 1034, 1036, 1038, 1041
\textbraceleft . 1031, 1034, 1038, 1041
\textbraceright 1031, 1034, 1038, 1041
\texteuro 1076
\texttt 816, 1037, 1094
\the . . 77, 78, 79, 80, 81, 82, 83, 84, 97, 509, 511, 517, 519, 526, 528, 534, 536, 543, 545, 551, 553, 562, 574, 587, 599, 612, 624, 635, 644, 692, 712, 713, 821, 832, 934, 940, 941, 942, 943, 944, 950, 951, 952, 975, 981, 990
\thr@@ 201, 286, 415, 446
\TMP@EnsureCode 94, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127
\tracingonline 931
\tw@ 203, 205, 207, 288, 290, 292, 413, 448

U

\UNDEFINED 754, 1100
\unexpanded 753, 754, 801
\unskip 245, 330
\usepackage 757, 758, 759, 760, 762, 763, 764, 765, 800, 803, 808, 885

V

\v 1081
\vbox 956
\vrule 1003, 1014

W

\wd 934, 935, 942, 943, 969, 970, 975, 981, 990
\write 23, 52

X

\x 14, 15, 18, 22, 26, 28, 51, 56, 66, 75, 87, 399, 409, 423, 435, 509, 511, 517, 519, 526, 528, 534, 536, 543, 545, 551, 553, 560, 566, 572, 579, 585, 591, 597, 604, 610, 616, 622, 629, 633, 639, 642, 648, 1054, 1057

Z

\z@ 419, 438, 513, 521, 530, 538, 547, 555, 828, 829