The soulutf8 package

Heiko Oberdiek[*]

2019/12/15 v1.2

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

[*] Please report any issues at https://github.com/ho-tex/soul/issues

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

```
⟨*package⟩
```

2.1 Reload check and package identification

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

```
\begingroup\catcode61\catcode48\catcode32=10\relax%
  \catcode13=5 % ^^M
  \endlinechar=13 %
  \catcode35=6 % #
  \catcode39=12 % '
  \catcode44=12 % ,
  \catcode45=12 % -
  \catcode46=12 % .
  \catcode58=12 % :
```

```
  \catcode64=11 % @
  \catcode123=1 % {
  \catcode125=2 % }
  \expandafter\let\expandafter\x\csname ver@soulutf8.sty\endcsname
  \ifx\x\relax % plain-TeX, first loading
  \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{soulutf8}{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@soulutf8.sty\endcsname
\ProvidesPackage{soulutf8}%
```

```
  [2019/12/15 v1.2 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}[2019/12/03]%
  \RequirePackage{etexcmds}[2019/12/15]%
\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 Installation

3.1 Download

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

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

CTAN:macros/latex/contrib/soulutf8/soulutf8.pdf Documentation.

[1] CTAN:pkg/soulutf8

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/soulutf8.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 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/soulutf8/soulutf8.sty
soulutf8.pdf → doc/latex/soulutf8/soulutf8.pdf
soulutf8.dtx → source/latex/soulutf8/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.

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.

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
```

4 References

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

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

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

5 History

[2007/09/09 v1.0]

- First version.

[2016/05/16 v1.1]

- Documentation updates.

[2019/12/15 v1.2]

- Documentation updates.

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