

The `soulutf8` package

Heiko Oberdiek*

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	2
2.1	Reload check and package identification	2
2.2	Catcodes	4
2.3	Loading packages	5
2.3.1	plain $\mathrm{T}_{\mathrm{E}}\mathrm{X}$	5
2.3.2	$\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$	6
2.3.3	ε - $\mathrm{T}_{\mathrm{E}}\mathrm{X}$	6
2.4	Macro for redefinitions	6
2.5	Redefinition of <code>\SOUL@eval</code>	6
2.6	UTF-8 analysis	10
2.6.1	Help strings	10
2.6.2	Support for <code>utf8.def</code>	10
2.6.3	Support for <code>utf8x.def</code>	11
2.7	Actions for UTF-8 sequences	11
2.7.1	Redefinition of <code>\SOUL@splittoken</code>	12
2.8	Patches	12
3	Installation	15
3.1	Download	15
3.2	Bundle installation	16
3.3	Package installation	16
3.4	Refresh file name databases	16
3.5	Some details for the interested	16
4	References	17
5	History	17
	[2007/09/09 v1.0]	17
	[2016/05/16 v1.1]	17

*Please report any issues at <https://github.com/ho-tex/oberdiek/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 implicate kernings are detected. However, the result may be incompatible with the original behaviour of package `soul` because of these respected implicate kernings.
- ε -T_EX, 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 L^AT_EX.

```
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
52 \immediate\write-1{Package: #3 #4}%
53 \xdef#1{#4}%
54 }%
55 \else
56 \def\x#1#2[#3]{\endgroup
57 #2[#3]}%
58 \ifx#1\@undefined
59 \xdef#1{#3}%
60 \fi
61 \ifx#1\relax
62 \xdef#1{#3}%
63 \fi
64 }%
65 \fi
66 \expandafter\x\csname ver@soulutf8.sty\endcsname
67 \ProvidesPackage{soulutf8}%

```

2.2 Catcodes

```

69 \begingroup\catcode61\catcode48\catcode32=10\relax%
70 \catcode13=5 % ^M
71 \endlinechar=13 %
72 \catcode123=1 % {
73 \catcode125=2 % }
74 \catcode64=11 % @
75 \def\x{\endgroup
76   \expandafter\edef\csname SOuL@AtEnd\endcsname{%
77     \endlinechar=\the\endlinechar\relax
78     \catcode13=\the\catcode13\relax
79     \catcode32=\the\catcode32\relax
80     \catcode35=\the\catcode35\relax
81     \catcode61=\the\catcode61\relax
82     \catcode64=\the\catcode64\relax
83     \catcode123=\the\catcode123\relax
84     \catcode125=\the\catcode125\relax
85   }%
86 }%
87 \x\catcode61\catcode48\catcode32=10\relax%
88 \catcode13=5 % ^M
89 \endlinechar=13 %
90 \catcode35=6 % #
91 \catcode64=11 % @
92 \catcode123=1 % {
93 \catcode125=2 % }
94 \def\TMP@EnsureCode#1#2{%
95   \edef\SOuL@AtEnd{%
96     \SOuL@AtEnd
97     \catcode#1=\the\catcode#1\relax
98   }%
99   \catcode#1=#2\relax
100 }
101 \TMP@EnsureCode{10}{12}% ^^J
102 \TMP@EnsureCode{33}{12}% !
103 \TMP@EnsureCode{34}{12}% "
104 \TMP@EnsureCode{36}{3}% $
105 \TMP@EnsureCode{39}{12}% '
106 \TMP@EnsureCode{40}{12}% (
107 \TMP@EnsureCode{41}{12}% )
108 \TMP@EnsureCode{42}{12}% *
109 \TMP@EnsureCode{43}{12}% +
110 \TMP@EnsureCode{44}{12}% ,
111 \TMP@EnsureCode{45}{12}% -
112 \TMP@EnsureCode{46}{12}% .
113 \TMP@EnsureCode{47}{12}% /
114 \TMP@EnsureCode{58}{12}% :
115 \TMP@EnsureCode{60}{12}% <
116 \TMP@EnsureCode{62}{12}% >
117 \TMP@EnsureCode{91}{12}% [
118 \TMP@EnsureCode{93}{12}% ]
119 \TMP@EnsureCode{94}{7}% ^
120 \TMP@EnsureCode{96}{12}% ‘
121 \TMP@EnsureCode{126}\active % ~
122 \TMP@EnsureCode{128}{12}% ^^80

```

```

123 \TMP@EnsureCode{147}{12}% ^^93
124 \TMP@EnsureCode{148}{12}% ^^94
125 \TMP@EnsureCode{160}{12}% ^^a0
126 \TMP@EnsureCode{194}{12}% ^^c2
127 \TMP@EnsureCode{226}{12}% ^^e2
128 \edef\Soul@AtEnd{\Soul@AtEnd\noexpand\endinput}

```

2.3 Loading packages

Package soul uses \documentclass to detect L^AT_EX.

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

2.3.1 plain T_EX

First we check, whether package soul is already loaded.

```
130 \expandafter\ifx\csname Soul@\endcsname\relax
```

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

```

131 \let\Soul@orgDeclareRobustCommand\DeclareRobustCommand
132 \let\Soul@orgnewcommand\newcommand
133 \let\Soul@orgDeclareOption\DeclareOption
134 \let\Soul@orgPackageError\PackageError
135 \def\Soul@restorelatexcmds{%
136 \let\DeclareRobustCommand\Soul@orgDeclareRobustCommand
137 \let\newcommand\Soul@orgnewcommand
138 \let\DeclareOption\Soul@orgDeclareOption
139 \let\PackageError\Soul@orgPackageError
140 }%
141 \input soul.sty\relax
142 \Soul@restorelatexcmds
143 \fi

```

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

```

144 \input infwarerr.sty\relax
145 \let\Soul@orgSOUL@error\Soul@error
146 \def\Soul@error{%
147 \begingroup
148 \let\PackageError\@PackageError
149 \Soul@orgSOUL@error
150 \endgroup
151 }%
152 \input etexcmds.sty\relax

```

\@onelevel@sanitize Define L^AT_EX's \@onelevel@sanitize if not already available.

```

153 \expandafter\ifx\csname @onelevel@sanitize\endcsname\relax
154 \def\@onelevel@sanitize#1{%
155 \edef#1{%
156 \expandafter\strip@prefix\meaning#1%
157 }%
158 }%

```

\strip@prefix

```

159 \def\strip@prefix#1>{}%
160 \fi
161 \else

```

2.3.2 L^AT_EX

```

162 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{soul}}%
163 \ProcessOptions\relax
164 \RequirePackage{soul}[2003/11/17]%
165 \RequirePackage{infwarerr}[2016/05/16]%
166 \RequirePackage{etexcmds}[2016/05/16]%
167 \fi

```

2.3.3 ε-T_EX

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

```

168 \ifetex@unexpanded
169 \catcode33=14 % '!' : comment
170 \catcode43=9 % '+' : ignore
171 \else
172 \catcode33=9 % '!' : ignore
173 \catcode43=14 % '+' : comment
174 \fi

```

2.4 Macro for redefinitions

\SOuL@redefine

```

175 \def\SOuL@redefine#1{%
176 \begingroup
177 \def\SOuL@cmd{#1}%
178 \afterassignment\SOuL@cmdcheck
179 \def\SOuL@temp
180 }

```

\SOuL@cmdcheck

```

181 \def\SOuL@cmdcheck{%
182 \expandafter\ifx\SOuL@cmd\SOuL@temp
183 \else
184 \edef\SOuL@temp*{\expandafter\string\SOuL@cmd}%
185 \@PackageWarningNoLine{soulutf8}{%
186 Command \SOuL@temp* has changed.\MessageBreak
187 Supported versions of package 'soul': 2003/11/17.\MessageBreak
188 Depending on the unknown changes the redefinition\MessageBreak
189 of \SOuL@temp* may not behave correctly%
190 }%
191 \fi
192 \expandafter\endgroup
193 \expandafter\def\SOuL@cmd
194 }

```

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.

```

195 \SOuL@redefine\SOUL@eval{%
First the expected definition.
196 \def\SOUL@n##1{\SOUL@scan}%
197 \if\noexpand\SOUL@@\SOUL@spc
198 \else

```

```

199 \SOUL@ignorespacesfalse
200 \fi
201 \ifnum\SOUL@minus=\thr@@
202 \SOUL@flushminus
203 \else\ifnum\SOUL@comma=\tw@
204 \SOUL@flushcomma
205 \else\ifnum\SOUL@apo=\tw@
206 \SOUL@flushapo
207 \else\ifnum\SOUL@grave=\tw@
208 \SOUL@flushgrave
209 \fi\fi\fi\fi
210 \ifx\SOUL@@-\else\SOUL@flushminus\fi
211 \ifx\SOUL@@,\else\SOUL@flushcomma\fi
212 \ifx\SOUL@@'\else\SOUL@flushapo\fi
213 \ifx\SOUL@@'\else\SOUL@flushgrave\fi
214 \ifx\SOUL@@-%
215 \advance\SOUL@minus\@ne
216 \else\ifx\SOUL@@,%
217 \advance\SOUL@comma\@ne
218 \else\ifx\SOUL@@'%
219 \advance\SOUL@apo\@ne
220 \else\ifx\SOUL@@'\%
221 \advance\SOUL@grave\@ne
222 \else
223 \SOUL@flushminus
224 \SOUL@flushcomma
225 \SOUL@flushapo
226 \SOUL@flushgrave
227 \ifx\SOUL@@\SOUL@stop
228 \def\SOUL@n*{%
229 \SOUL@doword
230 \SOUL@eventuallyexhyphen\null
231 }%
232 \else\ifx\SOUL@@\par
233 \def\SOUL@n*\par{\par\leavevmode\SOUL@scan}%
234 \else\if\noexpand\SOUL@@\SOUL@spc
235 \SOUL@doword
236 \SOUL@eventuallyexhyphen\null
237 \ifSOUL@ignorespaces
238 \else
239 \SOUL@everySPACE{}%
240 \fi
241 \def\SOUL@n* {\SOUL@scan}%
242 \else\ifx\SOUL@@\%
243 \SOUL@doword
244 \SOUL@eventuallyexhyphen\null
245 \SOUL@everySPACE{\unskip\nobreak\hfil\break}%
246 \SOUL@ignoreSPACEtrue
247 \else\ifx\SOUL@@~%
248 \SOUL@doword
249 \SOUL@eventuallyexhyphen\null
250 \SOUL@everySPACE{\nobreak}%
251 \else\ifx\SOUL@@\slash
252 \SOUL@doword
253 \SOUL@eventuallyexhyphen{/}%
254 \SOUL@exhyphen{/}%
255 \else\ifx\SOUL@@\mbox
256 \def\SOUL@n*{\SOUL@addprotect}%

```

[illegible]

```

281 \def\SOUL@n*##1{\SOUL@scan}%
282 \if\noexpand\SOUL@@\SOUL@spc
283 \else
284 \SOUL@ignorespacesfalse
285 \fi
286 \ifnum\SOUL@minus=\thr@@
287 \SOUL@flushminus
288 \else\ifnum\SOUL@comma=\tw@
289 \SOUL@flushcomma
290 \else\ifnum\SOUL@apo=\tw@
291 \SOUL@flushapo
292 \else\ifnum\SOUL@grave=\tw@
293 \SOUL@flushgrave
294 \fi\fi\fi\fi
295 \ifx\SOUL@@-\else\SOUL@flushminus\fi
296 \ifx\SOUL@@,\else\SOUL@flushcomma\fi
297 \ifx\SOUL@@'\else\SOUL@flushapo\fi
298 \ifx\SOUL@@'\else\SOUL@flushgrave\fi
299 \ifx\SOUL@@-%
300 \advance\SOUL@minus\@ne
301 \else\ifx\SOUL@@,%
302 \advance\SOUL@comma\@ne
303 \else\ifx\SOUL@@'
304 \advance\SOUL@apo\@ne
305 \else\ifx\SOUL@@'
306 \advance\SOUL@grave\@ne
307 \else
308 \SOUL@flushminus
309 \SOUL@flushcomma
310 \SOUL@flushapo
311 \SOUL@flushgrave
312 \ifx\SOUL@@\SOUL@stop
313 \def\SOUL@n*{%

```



```

314         \SOUL@doword
315         \SOUL@eventuallyexhyphen\null
316     }%
317 \else\ifx\SOUL@@\par
318     \def\SOUL@n*\par{\par\leavevmode\SOUL@scan}%
319 \else\if\noexpand\SOUL@@\SOUL@spc
320     \SOUL@doword
321     \SOUL@eventuallyexhyphen\null
322     \ifSOUL@ignorespaces
323     \else
324         \SOUL@everySPACE{}%
325     \fi
326     \def\SOUL@n* {\SOUL@scan}%
327 \else\ifx\SOUL@@\%
328     \SOUL@doword
329     \SOUL@eventuallyexhyphen\null
330     \SOUL@everySPACE{\unskip\nobreak\hfil\break}%
331     \SOUL@ignorespacestrue
332 \else\ifx\SOUL@@~%
333     \SOUL@doword
334     \SOUL@eventuallyexhyphen\null
335     \SOUL@everySPACE{\nobreak}%
336 \else\ifx\SOUL@@\slash
337     \SOUL@doword
338     \SOUL@eventuallyexhyphen{/}%
339     \SOUL@exhyphen{/}%
340 \else\ifx\SOUL@@\mbox
341     \def\SOUL@n*{\SOUL@addprotect}%
342 \else\ifx\SOUL@@\hbox
343     \def\SOUL@n*{\SOUL@addprotect}%
344 \else\ifx\SOUL@@\soulomit
345     \def\SOUL@n*\soulomit##1{%
346         \SOUL@doword
347         {\spaceskip\SOUL@spaceskip##1}%
348         \SOUL@scan
349     }%
350 \else\ifx\SOUL@@\break
351     \SOUL@doword
352     \break
353 \else\ifx\SOUL@@\linebreak
354     \SOUL@doword
355     \SOUL@everySPACE{\linebreak}%
356 \else\ifcat\bgroup\noexpand\SOUL@@
357     \def\SOUL@n*{\SOUL@addgroup{}}%
358 \else\ifcat$\noexpand\SOUL@@
359     \def\SOUL@n*{\SOUL@addmath}%
360 \else

```

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

```

361     \SOUL@analyzeutfviii
362     \ifcase\SOUL@octets
363         \SOUL@analyzeutfviiix
364     \fi
365     \ifcase\SOUL@octets
366         \def\SOUL@n*{\SOUL@dotoken}%
367     \or % 1
368     \or % 2
369         \def\SOUL@n*{\SOUL@addtwooctets}%
370     \or % 3

```


421 }

2.6.3 Support for utf8x.def

\SOuL@analyzeutfviiiix

```
422 \begingroup
423 \edef\x{\endgroup
424 \def\noexpand\SOuL@analyzeutfviiiix{%
425 \noexpand\expandafter\noexpand\SOuL@checkutfviiiix
426 \noexpand\meaning\noexpand\SOUL@@
427 \SOuL@stringmacrocolon\SOuL@charhash1{}{}{}{}%
428 \SOuL@stringcsnameu\SOuL@stringundeferr
429 \noexpand\@nil
430 }%
```

\SOuL@checkutfviiiix

```
431 \def\noexpand\SOuL@checkutfviiiix
432 ##1\SOuL@stringmacrocolon\SOuL@charhash1##2##3##4##5##6%
433 \SOuL@stringcsnameu##7\SOuL@stringundeferr##8\noexpand\@nil
434 }%
435 \x{%
436 \def\SOuL@temp{#7}%
437 \ifx\SOuL@temp\SOuL@empty
438 \chardef\SOuL@octets=\z@
439 \else
440 \def\SOuL@temp{#5}%
441 \ifx\SOuL@temp\SOuL@charthree
442 \chardef\SOuL@octets=4 %
443 \else
444 \def\SOuL@temp{#3}%
445 \ifx\SOuL@temp\SOuL@chartwo
446 \chardef\SOuL@octets=\thr@@
447 \else
448 \chardef\SOuL@octets=\tw@
449 \fi
450 \fi
451 \fi
452 }
```

2.7 Actions for UTF-8 sequences

\SOuL@addtwooctets

```
453 \def\SOuL@addtwooctets#1#2{%
454 \def\SOuL@temp{#1#2}%
455 \@onelevel@sanitize\SOuL@temp
456 \ifx\SOuL@temp\SOuL@stringnobreakspace
457 \SOUL@doword
458 \SOUL@eventuallyexhyphen\null
459 \SOUL@everyspace{\nobreak}%
460 \let\SOuL@next\SOUL@scan
461 \else
462 \def\SOuL@next{%
463 ! \SOUL@addtoken{\noexpand#1\noexpand#2}}%
464 + \SOUL@addtoken{\etex@unexpanded{#1#2}}}%
465 }%
466 \fi
467 \SOuL@next
468 }
```

\SOuL@addthreeoctets

```

469 \def\SOuL@addthreeoctets#1#2#3{%
470   \def\SOuL@temp{#1#2#3}%
471   \@onelevel@sanitize\SOuL@temp
472   \ifx\SOuL@temp\SOuL@stringendash
473     \SOUL@doword
474     \SOUL@eventuallyexhyphen{-}%
475     \SOUL@exhyphen{--}%
476     \let\SOuL@next\SOUL@scan
477   \else
478     \ifx\SOuL@temp\SOuL@stringemdash
479       \SOUL@doword
480       \SOUL@eventuallyexhyphen{-}%
481       \SOUL@exhyphen{---}%
482       \let\SOuL@next\SOUL@scan
483     \else
484       \def\SOuL@next{%
485         ! \SOUL@addtoken{\noexpand#1\noexpand#2\noexpand#3}}%
486       + \SOUL@addtoken{\etex@unexpanded{#1#2#3}}}%
487       }%
488     \fi
489   \fi
490   \SOuL@next
491 }

```

\SOuL@addfouroctets

```

492 \def\SOuL@addfouroctets#1#2#3#4{%
493   ! \SOUL@addtoken{\noexpand#1\noexpand#2\noexpand#3\noexpand#4}}%
494   + \SOUL@addtoken{\etex@unexpanded{#1#2#3#4}}}%
495 }

```

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.

```

496 \SOuL@redefine\SOUL@splittoken#1#2\SOUL@stop{%
497   \global\SOUL@token={#1}%
498   \global\SOUL@word={#2}%
499 }#1{%
500   \global\SOUL@token={#1}%
501   \SOuL@remainingtoken\relax
502 }

```

\SOuL@remainingtoken

```

503 \def\SOuL@remainingtoken#1\SOUL@stop{%
504   \global\SOUL@word=\expandafter{\@gobble#1}%
505 }

```

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 implicate 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

```
506 \SOuL@redefine\SOUL@flushcomma{%
507   \ifcase\SOUL@comma
508   \or
509     \edef\x{\SOUL@word={\the\SOUL@word,}}\x
510   \or
511     \edef\x{\SOUL@word={\the\SOUL@word{,,}}}\x
512   \fi
513   \SOUL@comma\z@
514 }{%
515   \ifcase\SOUL@comma
516   \or
517     \edef\x{\SOUL@word={\the\SOUL@word,}}\x
518   \or
519     \edef\x{\SOUL@word={\the\SOUL@word{,,}}}\x
520   \fi
521   \SOUL@comma\z@
522 }
```

\SOUL@flushapo

```
523 \SOuL@redefine\SOUL@flushapo{%
524   \ifcase\SOUL@apo
525   \or
526     \edef\x{\SOUL@word={\the\SOUL@word'}}\x
527   \or
528     \edef\x{\SOUL@word={\the\SOUL@word{' '}}}\x
529   \fi
530   \SOUL@apo\z@
531 }{%
532   \ifcase\SOUL@apo
533   \or
534     \edef\x{\SOUL@word={\the\SOUL@word'}}\x
535   \or
536     \edef\x{\SOUL@word={\the\SOUL@word{' '}}}\x
537   \fi
538   \SOUL@apo\z@
539 }
```

\SOUL@flushgrave

```
540 \SOuL@redefine\SOUL@flushgrave{%
541   \ifcase\SOUL@grave
542   \or
543     \edef\x{\SOUL@word={\the\SOUL@word'}}\x
544   \or
545     \edef\x{\SOUL@word={\the\SOUL@word{' '}}}\x
546   \fi
547   \SOUL@grave\z@
548 }{%
549   \ifcase\SOUL@grave
550   \or
551     \edef\x{\SOUL@word={\the\SOUL@word'}}\x
552   \or
553     \edef\x{\SOUL@word={\the\SOUL@word{' '}}}\x
554   \fi
555   \SOUL@grave\z@
556 }
```

\SOUL@addgroup

```

557 \SOuL@redefine\SOUL@addgroup#1#2{%
558   {%
559     \let\protect\noexpand
560     \edef\x{%
561       \global\SOUL@word={%
562         \the\SOUL@word
563         {\noexpand#1#2}}%
564     }%
565   }%
566   \x
567 }%
568 \SOUL@scan
569 }#1#2{%
570   \begingroup
571   \let\protect\noexpand
572   \edef\x{\endgroup
573     \SOUL@word={%
574       \the\SOUL@word
575       {\noexpand#1{#2}}}%
576   +   {\etex@unexpanded{#1{#2}}}%
577   }%
578 }%
579 \x
580 \SOUL@scan
581 }

```

\SOUL@addmath

```

582 \SOuL@redefine\SOUL@addmath$#1${%
583   {%
584     \let\protect\noexpand
585     \edef\x{%
586       \global\SOUL@word={%
587         \the\SOUL@word
588         {\hbox{$#1$}}}%
589     }%
590   }%
591   \x
592 }%
593 \SOUL@scan
594 }$#1${%
595   \begingroup
596   \let\protect\noexpand
597   \edef\x{\endgroup
598     \SOUL@word={%
599       \the\SOUL@word
600       {\hbox{$#1$}}}%
601   +   {\etex@unexpanded{\hbox{$#1$}}}%
602   }%
603 }%
604 \x
605 \SOUL@scan
606 }

```

\SOUL@addprotect

```

607 \SOuL@redefine\SOUL@addprotect#1#2{%
608   {%
609     \let\protect\noexpand
610     \edef\x{%

```

```

611     \global\SOUL@word={%
612     \the\SOUL@word
613     {\hbox{#2}}}%
614     }%
615   }%
616   \x
617 }%
618 \SOUL@scan
619 }#1#2{%
620 \begingroup
621 \let\protect\noexpand
622 \edef\x{\endgroup
623   \SOUL@word={%
624   \the\SOUL@word
625   !   {\hbox{#2}}}%
626 +   {\etex@unexpanded{\hbox{#2}}}%
627   }%
628 }%
629 \x
630 \SOUL@scan
631 }

\SOUL@addtoken

632 + \SOUL@redefine\SOUL@addtoken#1{%
633 +   \edef\x{%
634 +     \SOUL@word={%
635 +       \the\SOUL@word
636 +       \noexpand#1%
637 +     }%
638 +   }%
639 +   \x
640 +   \SOUL@scan
641 + }#1{%
642 +   \edef\x{%
643 +     \SOUL@word={%
644 +       \the\SOUL@word
645 +       \etex@unexpanded{#1}%
646 +     }%
647 +   }%
648 +   \x
649 +   \SOUL@scan
650 + }%

651 \SOUL@AtEnd%
652 </package>

```

3 Installation

3.1 Download

Package. This package is available on CTAN¹:

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

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

¹CTAN: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/oberdiek.tds.zip](#)

TDS refers to the standard “A Directory Structure for T_EX 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 T_EX:

```
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
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`.

3.4 Refresh file name databases

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

3.5 Some details for the interested

Unpacking with L^AT_EX. The `.dtx` chooses its action depending on the format:

plain T_EX: Run `docstrip` and extract the files.

L^AT_EX: Generate the documentation.

If you insist on using L^AT_EX for `docstrip` (really, `docstrip` does not need L^AT_EX), 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 pdfL^AT_EX:

```
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.

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.

Symbols	
\@PackageError	148
\@PackageWarningNoLine	185
\@gobble	504
\@ne	215, 217, 219, 221, 300, 302, 304, 306
\@nil	404, 407, 429, 433
\@onelevel@sanitiz	<u>153</u> , 381, 455, 471
\@undefined	58, 129
\\	242, 327
A	
\active	121
B	
\advance	215, 217, 219, 221, 300, 302, 304, 306
\afterassignment	178
\aftergroup	29
C	
\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	\null 230, 236, 244, 249, 315, 321, 329, 334, 458
\chardef 411, 438, 442, 446, 448	
\csname 14, 21, 50, 66, 76, 130, 153, 380, 381	P
\CurrentOption 162	\PackageError 134, 139, 148
	\PackageInfo 26
D	\par 232, 233, 317, 318
\DeclareOption 133, 138, 162	\PassOptionsToPackage 162
\DeclareRobustCommand 131, 136	\ProcessOptions 163
\documentclass 129	\protect .. 559, 571, 584, 596, 609, 621
	\ProvidesPackage 19, 67
E	R
\empty 17, 18	\RequirePackage 164, 165, 166
\endcsname 14, 21, 50, 66, 76, 130, 153, 380, 381	
\endinput 29, 128	S
\endlinechar 4, 35, 71, 77, 89	\slash 251, 336
\etex@unexpanded 464, 486, 494, 576, 601, 626, 645	\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
H	\SOUL@addgroup 272, 357, 557
\hbox 257, 342, 588, 600, 601, 613, 625, 626	\SOUL@addmath 274, 359, 582
\hfil 245, 330	\SOUL@addprotect 256, 258, 341, 343, 607
	\SOUL@addthreeoctets 371, 469
I	\SOUL@addtoken 463, 464, 485, 486, 493, 494, 632
\if 197, 234, 282, 319	\SOUL@addtwooctets 369, 453
\ifcase 362, 365, 507, 515, 524, 532, 541, 549	\SOUL@analyzeutfviii 361, 398
\ifcat 271, 273, 356, 358	\SOUL@analyzeutfviiix 363, 422
\ifetex@unexpanded 168	\SOUL@apo 205, 219, 290, 304, 524, 530, 532, 538
\ifnum 201, 203, 205, 207, 286, 288, 290, 292	\SOUL@AtEnd 95, 96, 128, 651
\ifSOUL@ignorespaces 237, 322	\SOUL@charhash 394, 427, 432
\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	\SOUL@charthree 396, 441
\immediate 23, 52	\SOUL@chartwo 395, 445
\input 141, 144, 152	\SOUL@checkutfviii 401, 406
	\SOUL@checkutfviiix 425, 431
L	\SOUL@cmd 177, 182, 184, 193
\leavevmode 233, 318	\SOUL@cmdcheck 178, 181
\linebreak 268, 270, 353, 355	\SOUL@comma 203, 217, 288, 302, 507, 513, 515, 521
	\SOUL@defsanitizedstring 379, 383, 384, 385, 386, 387, 388, 389, 390
M	\SOUL@dotoken 276, 366
\mbox 255, 340	\SOUL@doword 229, 235, 243, 248, 252, 261, 266, 269, 314, 320, 328, 333, 337, 346, 351, 354, 457, 473, 479
\meaning 156, 402, 426	\SOUL@empty 397, 437
\MessageBreak 186, 187, 188	\SOUL@error 144
	\SOUL@eval 195
N	
\newcommand 132, 137	
\nobreak 245, 250, 330, 335, 459	

<code>\SOUL@eventuallyexhyphen</code>	<code>\SOuL@stringnobreakspace</code> . . . 393, 456
. 230, 236, 244, 249, 253, 315,	<code>\SOuL@stringoctets</code> 403, 407
321, 329, 334, 338, 458, 474, 480	<code>\SOuL@stringthree</code> 414
<code>\SOUL@everySPACE</code> 239, 245,	<code>\SOuL@stringtwo</code> 412
250, 270, 324, 330, 335, 355, 459	<code>\SOuL@stringundeferr</code> 428, 433
<code>\SOUL@exhyphen</code> 254, 339, 475, 481	<code>\SOuL@stringUTFviii</code> 403, 407
<code>\SOUL@flushapo</code>	<code>\SOuL@temp</code> 179, 182, 184,
. 206, 212, 225, 291, 297, 310, 523	186, 189, 410, 412, 414, 416,
<code>\SOUL@flushcomma</code>	436, 437, 440, 441, 444, 445,
. 204, 211, 224, 289, 296, 309, 506	454, 455, 456, 470, 471, 472, 478
<code>\SOUL@flushgrave</code>	<code>\SOUL@token</code> 497, 500
. 208, 213, 226, 293, 298, 311, 540	<code>\SOUL@word</code> 498, 504, 509, 511, 517,
<code>\SOUL@flushminus</code>	519, 526, 528, 534, 536, 543,
. 202, 210, 223, 287, 295, 308	545, 551, 553, 561, 562, 573,
<code>\SOUL@grave</code> 207,	574, 586, 587, 598, 599, 611,
221, 292, 306, 541, 547, 549, 555	612, 623, 624, 634, 635, 643, 644
<code>\SOUL@ignorespacesfalse</code> . . . 199, 284	<code>\soulomit</code> 259, 260, 344, 345
<code>\SOUL@ignorespacestrue</code> 246, 331	<code>\spaceskip</code> 262, 347
<code>\SOUL@minus</code> 201, 215, 286, 300	<code>\strip@prefix</code> 156, 159
<code>\SOUL@n</code> 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	
<code>\SOuL@next</code>	
. 460, 462, 467, 476, 482, 484, 490	
<code>\SOuL@octets</code>	
. 362, 365, 411, 438, 442, 446, 448	
<code>\SOuL@orgDeclareOption</code> 133, 138	
<code>\SOuL@orgDeclareRobustCommand</code> . .	
. 131, 136	
<code>\SOuL@orgnewcommand</code> 132, 137	
<code>\SOuL@orgPackageError</code> 134, 139	
<code>\SOuL@orgSOUL@error</code> 145, 149	
<code>\SOuL@redefine</code> 175, 195, 496,	
506, 523, 540, 557, 582, 607, 632	
<code>\SOuL@remainingtoken</code> 501, 503	
<code>\SOuL@restorelatexcmds</code> 135, 142	
<code>\SOUL@scan</code>	
. 196, 233, 241, 263, 281, 318,	
326, 348, 460, 476, 482, 568,	
580, 593, 605, 618, 630, 640, 649	
<code>\SOUL@spaceskip</code> 262, 347	
<code>\SOUL@spc</code> 197, 234, 282, 319	
<code>\SOUL@splittoken</code> 496	
<code>\SOUL@stop</code> 227, 312, 496, 503	
<code>\SOuL@stringcsnameu</code> 428, 433	
<code>\SOuL@stringemdash</code> 392, 478	
<code>\SOuL@stringendash</code> 391, 472	
<code>\SOuL@stringfour</code> 416	
<code>\SOuL@stringmacrocolon</code> 427, 432	
	T
	<code>\the</code> 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
	<code>\thr@@</code> 201, 286, 415, 446
	<code>\TMP@EnsureCode</code> 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
	<code>\tw@</code> 203, 205, 207, 288, 290, 292, 413, 448
	U
	<code>\unskip</code> 245, 330
	W
	<code>\write</code> 23, 52
	X
	<code>\x</code> 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
	Z
	<code>\z@</code> 419, 438, 513, 521, 530, 538, 547, 555