

# The soulutf8 package

Heiko Oberdiek  
<heiko.oberdiek at gmail.com>

2007/09/09 v1.0

## 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

<b>1</b>	<b>Documentation</b>	<b>2</b>
1.1	Patch	2
1.2	Future	2
<b>2</b>	<b>Implementation</b>	<b>2</b>
2.1	Reload check and package identification	2
2.2	Catcodes	3
2.3	Loading packages	4
2.3.1	plain-TeX	4
2.3.2	L <sup>A</sup> T <sub>E</sub> X	5
2.3.3	ε-TeX	5
2.4	Macro for redefinitions	5
2.5	Redefinition of <code>\SOUL@eval</code>	6
2.6	UTF-8 analysis	9
2.6.1	Help strings	9
2.6.2	Support for <code>utf8.def</code>	9
2.6.3	Support for <code>utf8x.def</code>	10
2.7	Actions for UTF-8 sequences	10
2.7.1	Redefinition of <code>\SOUL@splittoken</code>	11
2.8	Patches	12
<b>3</b>	<b>Test</b>	<b>14</b>
3.1	Catcode checks for loading	14
3.2	L <sup>A</sup> T <sub>E</sub> Xtests	16
3.2.1	Driver definitions	17
3.2.2	Test macros	18
3.2.3	Testing	21
<b>4</b>	<b>Installation</b>	<b>22</b>
4.1	Download	22
4.2	Bundle installation	22
4.3	Package installation	22
4.4	Refresh file name databases	23
4.5	Some details for the interested	23
<b>5</b>	<b>References</b>	<b>24</b>

<b>6 History</b>	<b>24</b>
[2007/09/09 v1.0] . . . . .	24
<b>7 Index</b>	<b>24</b>

## 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.
- $\varepsilon$ -T<sub>E</sub>X, 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<sup>A</sup>T<sub>E</sub>X.

```
2 \begingroup
3 \catcode44 12 % ,
4 \catcode45 12 % -
5 \catcode46 12 % .
6 \catcode58 12 % :
7 \catcode64 11 % @
8 \catcode123 1 % {
9 \catcode125 2 % }
10 \expandafter\let\expandafter\x\csname ver@soulutf8.sty\endcsname
11 \ifx\x\relax % plain-TeX, first loading
```

```

12 \else
13   \def\empty{}%
14   \ifx\x\empty % LaTeX, first loading,
15   % variable is initialized, but \ProvidesPackage not yet seen
16   \else
17     \catcode35 6 % #
18     \expandafter\ifx\csname PackageInfo\endcsname\relax
19     \def\x#1#2{%
20       \immediate\write-1{Package #1 Info: #2.}%
21     }%
22   \else
23     \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
24   \fi
25   \x{soulutf8}{The package is already loaded}%
26   \aftergroup\endinput
27 \fi
28 \fi
29 \endgroup

```

Package identification:

```

30 \begingroup
31 \catcode35 6 % #
32 \catcode40 12 % (
33 \catcode41 12 % )
34 \catcode44 12 % ,
35 \catcode45 12 % -
36 \catcode46 12 % .
37 \catcode47 12 % /
38 \catcode58 12 % :
39 \catcode64 11 % @
40 \catcode91 12 % [
41 \catcode93 12 % ]
42 \catcode123 1 % {
43 \catcode125 2 % }
44 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
45   \def\x#1#2#3[#4]{\endgroup
46     \immediate\write-1{Package: #3 #4}%
47     \xdef#1{#4}%
48   }%
49 \else
50   \def\x#1#2[#3]{\endgroup
51     #2[#{#3}]%
52     \ifx#1\@undefined
53       \xdef#1{#3}%
54     \fi
55     \ifx#1\relax
56       \xdef#1{#3}%
57     \fi
58   }%
59 \fi
60 \expandafter\x\csname ver@soulutf8.sty\endcsname
61 \ProvidesPackage{soulutf8}%
62 [2007/09/09 v1.0 Adding support for UTF-8 to soul (HO)]

```

## 2.2 Catcodes

```

63 \begingroup
64 \catcode123 1 % {
65 \catcode125 2 % }
66 \def\x{\endgroup
67   \expandafter\edef\csname SOul@AtEnd\endcsname{%
68     \catcode35 \the\catcode35\relax
69     \catcode64 \the\catcode64\relax

```

```

70     \catcode123 \the\catcode123\relax
71     \catcode125 \the\catcode125\relax
72   }%
73 }%
74 \x
75 \catcode35 6 % #
76 \catcode64 11 % @
77 \catcode123 1 % {
78 \catcode125 2 % }
79 \def\TMP@EnsureCode#1#2{%
80   \edef\Soul@AtEnd{%
81     \Soul@AtEnd
82     \catcode#1 \the\catcode#1\relax
83   }%
84   \catcode#1 #2\relax
85 }
86 \TMP@EnsureCode{10}{12}% ^^J
87 \TMP@EnsureCode{33}{12}% !
88 \TMP@EnsureCode{34}{12}% "
89 \TMP@EnsureCode{36}{3}% $
90 \TMP@EnsureCode{39}{12}% '
91 \TMP@EnsureCode{40}{12}% (
92 \TMP@EnsureCode{41}{12}% )
93 \TMP@EnsureCode{42}{12}% *
94 \TMP@EnsureCode{43}{12}% +
95 \TMP@EnsureCode{44}{12}% ,
96 \TMP@EnsureCode{45}{12}% -
97 \TMP@EnsureCode{46}{12}% .
98 \TMP@EnsureCode{47}{12}% /
99 \TMP@EnsureCode{58}{12}% :
100 \TMP@EnsureCode{60}{12}% <
101 \TMP@EnsureCode{61}{12}% =
102 \TMP@EnsureCode{62}{12}% >
103 \TMP@EnsureCode{94}{7}% ^
104 \TMP@EnsureCode{96}{12}% ‘
105 \TMP@EnsureCode{126}\active % ~
106 \TMP@EnsureCode{128}{12}% ^^80
107 \TMP@EnsureCode{147}{12}% ^^93
108 \TMP@EnsureCode{148}{12}% ^^94
109 \TMP@EnsureCode{160}{12}% ^^a0
110 \TMP@EnsureCode{194}{12}% ^^c2
111 \TMP@EnsureCode{226}{12}% ^^e2

```

## 2.3 Loading packages

Package soul uses `\documentclass` to detect L<sup>A</sup>T<sub>E</sub>X.

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

### 2.3.1 plain-T<sub>E</sub>X

First we check, whether package soul is already loaded.

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

In case of plain-T<sub>E</sub>X 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.

```

114   \let\Soul@orgDeclareRobustCommand\DeclareRobustCommand
115   \let\Soul@orgnewcommand           \newcommand
116   \let\Soul@orgDeclareOption       \DeclareOption
117   \let\Soul@orgPackageError        \PackageError
118   \def\Soul@restorelatexcmds{%
119     \let\DeclareRobustCommand\Soul@orgDeclareRobustCommand
120     \let\newcommand           \Soul@orgnewcommand

```

```

121     \let\DeclareOption      \SOuL@orgDeclareOption
122     \let\PackageError      \SOuL@orgPackageError
123   }%
124   \input soul.sty\relax
125   \SOuL@restorelatexcmds
126   \fi

```

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

```

127   \input infwarerr.sty\relax
128   \let\SOuL@orgSOUL@error\SOUL@error
129   \def\SOUL@error{%
130     \begingroup
131     \let\PackageError\@PackageError
132     \SOuL@orgSOUL@error
133   \endgroup
134   }%
135   \input etexcmds.sty\relax

```

\@onelevel@sanitize Define L<sup>A</sup>T<sub>E</sub>X's \@onelevel@sanitize if not already available.

```

136   \expandafter\ifx\cename @onelevel@sanitize\endcename\relax
137   \def\@onelevel@sanitize#1{%
138     \edef#1{%
139       \expandafter\strip@prefix\meaning#1%
140     }%
141   }%

```

\strip@prefix

```

142   \def\strip@prefix#1>{-%}
143   \fi
144   \else

```

### 2.3.2 L<sup>A</sup>T<sub>E</sub>X

```

145   \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{soul}}%
146   \ProcessOptions\relax
147   \RequirePackage{soul}[2003/11/17]%
148   \RequirePackage{infwarerr}[2007/09/09]%
149   \RequirePackage{etexcmds}[2007/09/09]%
150   \fi

```

### 2.3.3 ε-<sub>T</sub>E<sub>X</sub>

In plain-<sub>T</sub>E<sub>X</sub> command \+ is an *outer* macro. Therefore numbers are used to avoid problems.

```

151   \ifetex@unexpanded
152     \catcode33=14 % '!' : comment
153     \catcode43=9  % '+' : ignore
154   \else
155     \catcode33=9  % '!' : ignore
156     \catcode43=14 % '+' : comment
157   \fi

```

## 2.4 Macro for redefinitions

\SOuL@redefine

```

158   \def\SOuL@redefine#1{%
159     \begingroup
160     \def\SOuL@cmd{#1}%
161     \afterassignment\SOuL@cmdcheck
162     \def\SOuL@temp
163   }

```

`\SOUL@cmdcheck`

```
164 \def\SOUL@cmdcheck{%
165   \expandafter\ifx\SOUL@cmd\SOUL@temp
166   \else
167     \edef\SOUL@temp*{\expandafter\string\SOUL@cmd}%
168     \@PackageWarningNoLine{soulutf8}{%
169       Command \SOUL@temp* has changed.\MessageBreak
170       Supported versions of package ‘soul’: 2003/11/17.\MessageBreak
171       Depending on the unknown changes the redefinition\MessageBreak
172       of \SOUL@temp* may not behave correctly%
173     }%
174   \fi
175 \expandafter\endgroup
176 \expandafter\def\SOUL@cmd
177 }
```

## 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.

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

First the expected definition.

```
179 \def\SOUL@n*##1{\SOUL@scan}%
180 \if\noexpand\SOUL@@\SOUL@spc
181 \else
182   \SOUL@ignorespacesfalse
183 \fi
184 \ifnum\SOUL@minus=\thr@@
185   \SOUL@flushminus
186 \else\ifnum\SOUL@comma=\tw@
187   \SOUL@flushcomma
188 \else\ifnum\SOUL@apo=\tw@
189   \SOUL@flushapo
190 \else\ifnum\SOUL@grave=\tw@
191   \SOUL@flushgrave
192 \fi\fi\fi\fi
193 \ifx\SOUL@@-\else\SOUL@flushminus\fi
194 \ifx\SOUL@@,\else\SOUL@flushcomma\fi
195 \ifx\SOUL@@'\else\SOUL@flushapo\fi
196 \ifx\SOUL@@'\else\SOUL@flushgrave\fi
197 \ifx\SOUL@@-%
198   \advance\SOUL@minus\@ne
199 \else\ifx\SOUL@@,%
200   \advance\SOUL@comma\@ne
201 \else\ifx\SOUL@@'%
202   \advance\SOUL@apo\@ne
203 \else\ifx\SOUL@@'\%
204   \advance\SOUL@grave\@ne
205 \else
206   \SOUL@flushminus
207   \SOUL@flushcomma
208   \SOUL@flushapo
209   \SOUL@flushgrave
210   \ifx\SOUL@@\SOUL@stop
211     \def\SOUL@n*{%
212       \SOUL@doword
213       \SOUL@eventuallyexhyphen\@null
214     }%
215   \else\ifx\SOUL@@\par
216     \def\SOUL@n*\par{\par\leavevmode\SOUL@scan}%

```

```

217 \else\if\noexpand\SOUL@@\SOUL@spc
218 \SOUL@doword
219 \SOUL@eventuallyexhyphen\null
220 \ifSOUL@ignorespaces
221 \else
222 \SOUL@everyspace{}%
223 \fi
224 \def\SOUL@n* {\SOUL@scan}%
225 \else\ifx\SOUL@@\%
226 \SOUL@doword
227 \SOUL@eventuallyexhyphen\null
228 \SOUL@everyspace{\unskip\nobreak\hfil\break}%
229 \SOUL@ignorespacestrue
230 \else\ifx\SOUL@@~%
231 \SOUL@doword
232 \SOUL@eventuallyexhyphen\null
233 \SOUL@everyspace{\nobreak}%
234 \else\ifx\SOUL@@\slash
235 \SOUL@doword
236 \SOUL@eventuallyexhyphen{/}%
237 \SOUL@exhyphen{/}%
238 \else\ifx\SOUL@@\mbox
239 \def\SOUL@n*{\SOUL@addprotect}%
240 \else\ifx\SOUL@@\hbox
241 \def\SOUL@n*{\SOUL@addprotect}%
242 \else\ifx\SOUL@@\soulomit
243 \def\SOUL@n*\soulomit##1{%
244 \SOUL@doword
245 {\spaceskip\SOUL@spaceskip##1}%
246 \SOUL@scan
247 }%
248 \else\ifx\SOUL@@\break
249 \SOUL@doword
250 \break
251 \else\ifx\SOUL@@\linebreak
252 \SOUL@doword
253 \SOUL@everyspace{\linebreak}%
254 \else\ifcat\bgroup\noexpand\SOUL@@
255 \def\SOUL@n*{\SOUL@addgroup{}}%
256 \else\ifcat$\noexpand\SOUL@@
257 \def\SOUL@n*{\SOUL@addmath}%
258 \else
259 \def\SOUL@n*{\SOUL@dotoken}%
260 \fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi
261 \fi\fi\fi\fi
262 \SOUL@n*%
263 }{%

```

Now the redefined version follows.

```

264 \def\SOUL@n*##1{\SOUL@scan}%
265 \if\noexpand\SOUL@@\SOUL@spc
266 \else
267 \SOUL@ignorespacesfalse
268 \fi
269 \ifnum\SOUL@minus=\thr@@
270 \SOUL@flushminus
271 \else\ifnum\SOUL@comma=\tw@
272 \SOUL@flushcomma
273 \else\ifnum\SOUL@apo=\tw@
274 \SOUL@flushapo
275 \else\ifnum\SOUL@grave=\tw@
276 \SOUL@flushgrave
277 \fi\fi\fi\fi

```

```

278 \ifx\SOUL@@-\else\SOUL@flushminus\fi
279 \ifx\SOUL@,@,\else\SOUL@flushcomma\fi
280 \ifx\SOUL@@'\else\SOUL@flushapo\fi
281 \ifx\SOUL@@'\else\SOUL@flushgrave\fi
282 \ifx\SOUL@@-%
283   \advance\SOUL@minus\@ne
284 \else\ifx\SOUL@,@,%
285   \advance\SOUL@comma\@ne
286 \else\ifx\SOUL@@'\%
287   \advance\SOUL@apo\@ne
288 \else\ifx\SOUL@@'\%
289   \advance\SOUL@grave\@ne
290 \else
291   \SOUL@flushminus
292   \SOUL@flushcomma
293   \SOUL@flushapo
294   \SOUL@flushgrave
295   \ifx\SOUL@@\SOUL@stop
296     \def\SOUL@n*{%
297       \SOUL@doword
298       \SOUL@eventuallyexhyphen\null
299     }%
300   \else\ifx\SOUL@@\par
301     \def\SOUL@n*\par{\par\leavevmode\SOUL@scan}%
302   \else\if\noexpand\SOUL@@\SOUL@spc
303     \SOUL@doword
304     \SOUL@eventuallyexhyphen\null
305     \ifSOUL@ignorespaces
306     \else
307       \SOUL@everyspace{%}%
308     \fi
309     \def\SOUL@n* {\SOUL@scan}%
310   \else\ifx\SOUL@@\%
311     \SOUL@doword
312     \SOUL@eventuallyexhyphen\null
313     \SOUL@everyspace{\unskip\nobreak\hfil\break}%
314     \SOUL@ignorespacestrue
315   \else\ifx\SOUL@@~%
316     \SOUL@doword
317     \SOUL@eventuallyexhyphen\null
318     \SOUL@everyspace{\nobreak}%
319   \else\ifx\SOUL@@\slash
320     \SOUL@doword
321     \SOUL@eventuallyexhyphen{/}%
322     \SOUL@exhyphen{/}%
323   \else\ifx\SOUL@@\mbox
324     \def\SOUL@n*{\SOUL@addprotect}%
325   \else\ifx\SOUL@@\hbox
326     \def\SOUL@n*{\SOUL@addprotect}%
327   \else\ifx\SOUL@@\soulomit
328     \def\SOUL@n*\soulomit##1{%
329       \SOUL@doword
330       {\spaceskip\SOUL@spaceskip##1}%
331       \SOUL@scan
332     }%
333   \else\ifx\SOUL@@\break
334     \SOUL@doword
335     \break
336   \else\ifx\SOUL@@\linebreak
337     \SOUL@doword
338     \SOUL@everyspace{\linebreak}%
339   \else\ifcat\bgroup\noexpand\SOUL@@

```



```

340     \def\SOUL@n*{\SOUL@addgroup{}}%
341     \else\ifcat$\noexpand\SOUL@@
342     \def\SOUL@n*{\SOUL@addmath}%
343     \else

```

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

```

344     \SOUL@analyzeutfviii
345     \ifcase\SOUL@octets
346     \SOUL@analyzeutfviiiix
347     \fi
348     \ifcase\SOUL@octets
349     \def\SOUL@n*{\SOUL@dotoken}%
350     \or % 1
351     \or % 2
352     \def\SOUL@n*{\SOUL@addtwooctets}%
353     \or % 3
354     \def\SOUL@n*{\SOUL@addthreeoctets}%
355     \or % 4
356     \def\SOUL@n*{\SOUL@addfouroctets}%
357     \fi
358     \fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi
359     \fi\fi\fi\fi
360     \SOUL@n*%
361 }

```

## 2.6 UTF-8 analysis

### 2.6.1 Help strings

```

362 \def\SOUL@defsanitizedstring#1#2{%
363   \expandafter\def\csname SOUL@string#1\endcsname{#2}%
364   \expandafter\@onelevel@sanitize\csname SOUL@string#1\endcsname
365 }
366 \SOUL@defsanitizedstring{UTFviii}{UTFviii@}
367 \SOUL@defsanitizedstring{octets}{@octets}
368 \SOUL@defsanitizedstring{two}{two}
369 \SOUL@defsanitizedstring{three}{three}
370 \SOUL@defsanitizedstring{four}{four}
371 \SOUL@defsanitizedstring{macrocolon}{macro:}
372 \SOUL@defsanitizedstring{csnameu}{csname u8-}
373 \SOUL@defsanitizedstring{undeferr}{utf@viii@undeferr}
374 \def\SOUL@stringendash{^^e2^^80^^93}
375 \def\SOUL@stringemdash{^^e2^^80^^94}
376 \def\SOUL@stringnobreakspace{^^c2^^a0}
377 \edef\SOUL@charhash{\string #}
378 \edef\SOUL@chartwo{\string 2}
379 \edef\SOUL@charthree{\string 3}
380 \def\SOUL@empty{}

```

### 2.6.2 Support for utf8.def

\SOUL@analyzeutfviii

```

381 \begingroup
382   \edef\x{\endgroup
383     \def\noexpand\SOUL@analyzeutfviii{%
384       \noexpand\expandafter\noexpand\SOUL@checkutfviii
385       \noexpand\meaning\noexpand\SOUL@@
386       \SOUL@stringUTFviii\SOUL@stringoctets
387       \noexpand\@nil
388     }%
389     \def\noexpand\SOUL@checkutfviii
390       ##1\SOUL@stringUTFviii##2\SOUL@stringoctets##3\noexpand\@nil
391   }%
392 \x{%

```

```

393 \def\Soul@temp{#2}%
394 \chardef\Soul@octets=%
395 \ifx\Soul@temp\Soul@stringtwo
396 \tw@
397 \else\ifx\Soul@temp\Soul@stringthree
398 \thr@@
399 \else\ifx\Soul@temp\Soul@stringfour
400 4 %
401 \else
402 \z@
403 \fi\fi\fi
404 }

```

### 2.6.3 Support for utf8x.def

\Soul@analyzeutfviiiix

```

405 \begingroup
406 \edef\x{\endgroup
407 \def\noexpand\Soul@analyzeutfviiiix{%
408 \noexpand\expandafter\noexpand\Soul@checkutfviiiix
409 \noexpand\meaning\noexpand\Soul@
410 \Soul@stringmacrocolon\Soul@charhash1{}-{}-{}-{}%
411 \Soul@stringcsnameu\Soul@stringundeferr
412 \noexpand\@nil
413 }%

```

\Soul@checkutfviiiix

```

414 \def\noexpand\Soul@checkutfviiiix
415 ##1\Soul@stringmacrocolon\Soul@charhash1##2##3##4##5##6%
416 \Soul@stringcsnameu##7\Soul@stringundeferr##8\noexpand\@nil
417 }%
418 \x{%
419 \def\Soul@temp{#7}%
420 \ifx\Soul@temp\Soul@empty
421 \chardef\Soul@octets=\z@
422 \else
423 \def\Soul@temp{#5}%
424 \ifx\Soul@temp\Soul@charthree
425 \chardef\Soul@octets=4 %
426 \else
427 \def\Soul@temp{#3}%
428 \ifx\Soul@temp\Soul@chartwo
429 \chardef\Soul@octets=\thr@@
430 \else
431 \chardef\Soul@octets=\tw@
432 \fi
433 \fi
434 \fi
435 }

```

## 2.7 Actions for UTF-8 sequences

\Soul@addtwooctets

```

436 \def\Soul@addtwooctets#1#2{%
437 \def\Soul@temp{#1#2}%
438 \@onelevel@sanitize\Soul@temp
439 \ifx\Soul@temp\Soul@stringnobreakspace
440 \Soul@doword
441 \Soul@eventuallyexhyphen\null
442 \Soul@everyspace{\nobreak}%
443 \let\Soul@next\Soul@scan

```

```

444 \else
445   \def\Soul@next{%
446 !     \Soul@addtoken{\noexpand#1\noexpand#2}}%
447 +     \Soul@addtoken{\etex@unexpanded{#1#2}}}%
448   }%
449 \fi
450 \Soul@next
451 }

```

`\Soul@addthreectets`

```

452 \def\Soul@addthreectets#1#2#3{%
453   \def\Soul@temp{#1#2#3}%
454   \@onelevel@sanitize\Soul@temp
455   \ifx\Soul@temp\Soul@stringendash
456     \Soul@doword
457     \Soul@eventuallyexhyphen{-}%
458     \Soul@exhyphen{--}%
459     \let\Soul@next\Soul@scan
460   \else
461     \ifx\Soul@temp\Soul@stringemdash
462       \Soul@doword
463       \Soul@eventuallyexhyphen{-}%
464       \Soul@exhyphen{---}%
465       \let\Soul@next\Soul@scan
466     \else
467       \def\Soul@next{%
468 !         \Soul@addtoken{\noexpand#1\noexpand#2\noexpand#3}}%
469 +         \Soul@addtoken{\etex@unexpanded{#1#2#3}}}%
470       }%
471     \fi
472   \fi
473   \Soul@next
474 }

```

`\Soul@addfourctets`

```

475 \def\Soul@addfourctets#1#2#3#4{%
476 ! \Soul@addtoken{\noexpand#1\noexpand#2\noexpand#3\noexpand#4}}%
477 + \Soul@addtoken{\etex@unexpanded{#1#2#3#4}}}%
478 }

```

### 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.

```

479 \Soul@redefine\Soul@splittoken#1#2\Soul@stop{%
480   \global\Soul@token={#1}%
481   \global\Soul@word={#2}%
482 }#1{%
483   \global\Soul@token={#1}%
484   \Soul@remainingtoken\relax
485 }

```

`\Soul@remainingtoken`

```

486 \def\Soul@remainingtoken#1\Soul@stop{%
487   \global\Soul@word=\expandafter{\@gobble#1}%
488 }

```

## 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`

```
489 \SOUL@redefine\SOUL@flushcomma{%
490   \ifcase\SOUL@comma
491   \or
492     \edef\x{\SOUL@word={\the\SOUL@word,}}\x
493   \or
494     \edef\x{\SOUL@word={\the\SOUL@word{,,}}}\x
495   \fi
496   \SOUL@comma\z@
497 }{%
498   \ifcase\SOUL@comma
499   \or
500     \edef\x{\SOUL@word={\the\SOUL@word,}}\x
501   \or
502     \edef\x{\SOUL@word={\the\SOUL@word{,,}}}\x
503   \fi
504   \SOUL@comma\z@
505 }
```

`\SOUL@flushapo`

```
506 \SOUL@redefine\SOUL@flushapo{%
507   \ifcase\SOUL@apo
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@apo\z@
514 }{%
515   \ifcase\SOUL@apo
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@apo\z@
522 }
```

`\SOUL@flushgrave`

```
523 \SOUL@redefine\SOUL@flushgrave{%
524   \ifcase\SOUL@grave
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@grave\z@
531 }{%
532   \ifcase\SOUL@grave
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@grave\z@
539 }
```

\SOUL@addgroup

```
540 \SOUL@redefine\SOUL@addgroup#1#2{%
541   {%
542     \let\protect\noexpand
543     \edef\x{%
544       \global\SOUL@word={%
545         \the\SOUL@word
546         {\noexpand#1#2}}%
547     }%
548   }%
549   \x
550 }%
551 \SOUL@scan
552 }#1#2{%
553   \beginingroup
554     \let\protect\noexpand
555     \edef\x{\endgroup
556       \SOUL@word={%
557         \the\SOUL@word
558 !       {\noexpand#1{#2}}%
559 +       {\etex@unexpanded{#1{#2}}}%
560     }%
561   }%
562   \x
563   \SOUL@scan
564 }
```

\SOUL@addmath

```
565 \SOUL@redefine\SOUL@addmath$#1${%
566   {%
567     \let\protect\noexpand
568     \edef\x{%
569       \global\SOUL@word={%
570         \the\SOUL@word
571         {\hbox{ $#1$ }}%
572     }%
573   }%
574   \x
575 }%
576 \SOUL@scan
577 }$#1${%
578   \beginingroup
579     \let\protect\noexpand
580     \edef\x{\endgroup
581       \SOUL@word={%
582         \the\SOUL@word
583 !       {\hbox{ $#1$ }}%
584 +       {\etex@unexpanded{\hbox{ $#1$ }}}%
585     }%
586   }%
587   \x
588   \SOUL@scan
589 }
```

\SOUL@addprotect

```
590 \SOUL@redefine\SOUL@addprotect#1#2{%
591   {%
592     \let\protect\noexpand
593     \edef\x{%
594       \global\SOUL@word={%
595         \the\SOUL@word
596         {\hbox{ #2 }}%
597     }%
598   }%
599   \x
600 }
```

```

597     }%
598   }%
599   \x
600 }%
601 \SOUL@scan
602 }#1#2{%
603 \begingroup
604   \let\protect\noexpand
605   \edef\x{\endgroup
606     \SOUL@word={%
607       \the\SOUL@word
608 !     {\hbox{#2}}}%
609 +     {\etex@unexpanded{\hbox{#2}}}}%
610   }%
611 }%
612 \x
613 \SOUL@scan
614 }

```

\SOUL@addtoken

```

615 + \SOuL@redefine\SOUL@addtoken#1{%
616 +   \edef\x{%
617 +     \SOUL@word={%
618 +       \the\SOUL@word
619 +       \noexpand#1%
620 +     }%
621 +   }%
622 +   \x
623 +   \SOUL@scan
624 + }#1{%
625 +   \edef\x{%
626 +     \SOUL@word={%
627 +       \the\SOUL@word
628 +       \etex@unexpanded{#1}%
629 +     }%
630 +   }%
631 +   \x
632 +   \SOUL@scan
633 + }%

634 \SOuL@AtEnd
635 </package>

```

## 3 Test

### 3.1 Catcode checks for loading

```

636 <*test1>
637 \catcode'\{=1 %
638 \catcode'\}=2 %
639 \catcode'\#=6 %
640 \catcode'\@=11 %
641 \expandafter\ifx\csname count@\endcsname\relax
642 \countdef\count@=255 %
643 \fi
644 \expandafter\ifx\csname @gobble\endcsname\relax
645 \long\def\@gobble#1{}%
646 \fi
647 \expandafter\ifx\csname @firstofone\endcsname\relax
648 \long\def\@firstofone#1{#1}%
649 \fi

```

```

650 \expandafter\ifx\csname loop\endcsname\relax
651 \expandafter\@firstofone
652 \else
653 \expandafter\@gobble
654 \fi
655 {%
656 \def\loop#1\repeat{%
657 \def\body{#1}%
658 \iterate
659 }%
660 \def\iterate{%
661 \body
662 \let\next\iterate
663 \else
664 \let\next\relax
665 \fi
666 \next
667 }%
668 \let\repeat=\fi
669 }%
670 \def\RestoreCatcodes{}
671 \count@=0 %
672 \loop
673 \edef\RestoreCatcodes{%
674 \RestoreCatcodes
675 \catcode\the\count@=\the\catcode\count@\relax
676 }%
677 \ifnum\count@<255 %
678 \advance\count@ 1 %
679 \repeat
680
681 \def\RangeCatcodeInvalid#1#2{%
682 \count@=#1\relax
683 \loop
684 \catcode\count@=15 %
685 \ifnum\count@<#2\relax
686 \advance\count@ 1 %
687 \repeat
688 }
689 \expandafter\ifx\csname LoadCommand\endcsname\relax
690 \def\LoadCommand{\input soulutf8.sty\relax}%
691 \fi
692 \def\Test{%
693 \RangeCatcodeInvalid{0}{47}%
694 \RangeCatcodeInvalid{58}{64}%
695 \RangeCatcodeInvalid{91}{96}%
696 \RangeCatcodeInvalid{123}{255}%
697 \catcode'\@=12 %
698 \catcode'\=0 %
699 \catcode'\{=1 %
700 \catcode'\}=2 %
701 \catcode'\#=6 %
702 \catcode'\[=12 %
703 \catcode'\]=12 %
704 \catcode'\%=14 %
705 \catcode'\ =10 %
706 \catcode13=5 %
707 \LoadCommand
708 \RestoreCatcodes
709 }
710 \Test
711 \csname @@end\endcsname

```

```
712 \end
713 </test1>
```

### 3.2 L<sup>A</sup>T<sub>E</sub>X tests

```
714 <*test2 | test3 | test4 | test5>
715 <noetex>\let\SavedUnexpanded\unexpanded
716 <noetex>\let\unexpanded\UNDEFINED
717 \NeedsTeXFormat{LaTeX2e}
718 \documentclass{article}
719 \usepackage[T1]{fontenc}
720 \usepackage{lmodern}
721 \usepackage{textcomp}
722 \usepackage[ngerman]{babel}
723 \expandafter\def\csname ver@hypdestopt.sty\endcsname{}
724 \usepackage[unicode]{hyperref}
725 \usepackage{bookmark}
726 \usepackage{color}
727 \usepackage{soul}[2003/11/17]
728 \makeatletter
729 \newcommand*{\@ifDefined}[1]{%
730   \@ifdefined{#1}{%
731     \@latex@error{%
732       \expandafter\noexpand\csname#1\endcsname undefined%
733     }\@ehc
734   }%
735 }
736 \newcommand*{\SaveOrg}[1]{%
737   \@ifdefinable{ORG@SOUL@#1}{%
738     \@ifDefined{SOUL@#1}{%
739       \expandafter\let\csname ORG@SOUL@#1\expandafter\endcsname
740         \csname SOUL@#1\endcsname
741     }%
742   }%
743 }
744 \newcommand*{\RestoreOrg}[1]{%
745   \@ifDefined{ORG@SOUL@#1}{%
746     \@ifDefined{SOUL@#1}{%
747       \expandafter\let\csname SOUL@#1\expandafter\endcsname
748         \csname ORG@SOUL@#1\endcsname
749     }%
750   }%
751 }
752 \makeatother
753 \SaveOrg{eval}
754 \SaveOrg{splittoken}
755 \SaveOrg{flushcomma}
756 \SaveOrg{flushapo}
757 \SaveOrg{grave}
758 \SaveOrg{addgroup}
759 \SaveOrg{addmath}
760 \SaveOrg{addprotect}
761 \SaveOrg{addtoken}
762 \usepackage{soulutf8}[2007/09/09]
763 <noetex>\let\unexpanded\SavedUnexpanded
```

Input encoding utf8.

```
764 <*utf8>
765 \usepackage[utf8]{inputenc}
766 \DeclareUnicodeCharacter{00A0}{\nobreakspace}
767 </utf8>
```

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



```

768 <*utf8x>
769 \let\origshow\show
770 \usepackage[utf8x]{inputenc}
771 \AtBeginDocument{%
772   \let\show\origshow
773 }
774 </utf8x>

```

### 3.2.1 Driver definitions

```
775 \makeatletter
```

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

```

776 \DeclareRobustCommand*\an}{%
777   \def\SOUL@preamble{${\^P}$}%
778   \def\SOUL@everyspace##1{##1\texttt{\char'\ }\linebreak[2]}%
779   \def\SOUL@postamble{${\^E}$}%
780   \def\SOUL@everyhyphen{${\^-}$\linebreak[1]}%
781   \def\SOUL@everyexhyphen##1{##1${\^=}}$%
782   \def\SOUL@everyyllable{${\^S}$}%
783   \def\SOUL@everytoken{\the\SOUL@token${\^T}$}%
784   \def\SOUL@everylowerthan{${\^L}$}%
785   \SOUL@
786 }

```

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

```

787 \DeclareRobustCommand*\sy}{%
788   \SOUL@setup
789   \def\SOUL@preamble{%
790     \lefthyphenmin=\z@
791     \righthyphenmin=\z@
792   }%
793   \def\SOUL@everyspace##1{##1\space}%
794   \def\SOUL@everyyllable{\the\SOUL@syllable}%
795   \def\SOUL@everyhyphen{%
796     \discretionary{%
797       \SOUL@setkern\SOUL@hyphkern
798       \SOUL@sethyphenchar
799     }{}{%
800       \hbox{\kern1pt$\cdot$}%
801     }%
802   }%
803   \def\SOUL@everyexhyphen##1{%
804     \SOUL@setkern\SOUL@hyphkern
805     \hbox{##1}%
806     \discretionary{}{}{%
807       \SOUL@setkern\SOUL@charkern
808     }%
809   }%
810   \SOUL@
811 }

```

\soulH

```

812 \DeclareRobustCommand*\soulH}{%
813   \def\SOUL@preamble{%
814     \gdef\SOULH{}%
815     \soulHadd{ }%
816   }%
817   \def\SOUL@everyspace##1{%
818     \soulHadd{Sp}%
819   }%
820   \def\SOUL@postamble{%
821     \soulHadd{]}%

```

```

822 }%
823 \def\SOUL@everyhyphen{%
824   \soulHadd{Hy}%
825 }%
826 \def\SOUL@everyexhyphen##1{%
827   \soulHadd{Exhy}%
828 }%
829 \def\SOUL@everysyllable{%
830   \soulHadd{Sy}%
831 }%
832 \def\SOUL@everytoken{%
833   \soulHadd{T}%
834 }%
835 \def\SOUL@everylowerthan{%
836   \soulHadd{<}%
837 }%
838 \SOUL@
839 }
840 \newcommand*{\soulHadd}[1]{%
841   #1%
842   \expandafter\gdef\expandafter\soulHstring\expandafter{%
843     \soulHstring
844     #1%
845   }%
846 }

```

### 3.2.2 Test macros

```

847 \usepackage{qstest}
848 \IncludeTests{*}
849 \LogTests{log}{*}{*}

850 \newsavebox\boxA
851 \newsavebox\boxB
852 \newsavebox\boxC

```

\TestDriver

```

853 \newcommand*{\TestDriver}[4]{%
854   \subsection{Driver #1}%
855   \@PackageInfo{soulutf8-test}{* Driver #1}%
856   \expandafter\let\expandafter\testdriver\csname #1\endcsname
857   \def\teststringA{#3}%
858   \def\teststringB{#4}%
859   \begin{qstest}{string-#2-driver-#1}{string,#2,driver,#1}%
860     \@PackageInfo{soulutf8-test}{* Version: soulutf8}%
861     \global\let\soulHstring\@empty
862     \setbox0=\hbox{\expandafter\testdriver\expandafter{\teststringA}}%
863     \let\HstringA\soulHstring
864     \SetTestVBox{\boxA}{\teststringA}%
865     %
866     \@PackageInfo{soulutf8-test}{* Version: soul + patch}%
867     \global\let\soulHstring\@empty
868     \RestoreOrg{eval}%
869     \setbox2=\hbox{\expandafter\testdriver\expandafter{\teststringB}}%
870     \let\HstringB\soulHstring
871     \SetTestVBox{\boxB}{\teststringB}%
872     %
873     \@PackageInfo{soulutf8-test}{* Version: original soul}%
874     \global\let\soulHstring\@empty
875     \RestoreOrg{splittoken}%
876     \RestoreOrg{flushcomma}%
877     \RestoreOrg{flushapo}%
878     \RestoreOrg{grave}%

```

```

879 \RestoreOrg{addgroup}%
880 \RestoreOrg{addmath}%
881 \RestoreOrg{addprotect}%
882 \RestoreOrg{addtoken}%
883 \setbox4=\hbox{\expandafter\testdriver\expandafter{\teststringB}}%
884 \let\HstringC\soulHstring
885 \SetTestVBox{\boxC}{\teststringB}%
886 %
887 \TestInfo{0}{soulutf8}%
888 \TestInfo{2}{soul + patch}%
889 \TestInfo{4}{original soul}%
890 \TestInfoVBox
891 \showboxdepth=\maxdimen
892 \showboxbreadth=\maxdimen
893 \tracingonline=@ne
894 \Expect*{\HstringA}*{\HstringB}%
895 \Expect*{\HstringA}*{\HstringC}%
896 \Expect*{\the\wd0}*{\the\wd2}%
897 \ifdim\wd0=\wd2 %
898 \else
899 \showbox0 %
900 \showbox2 %
901 \fi
902 \Expect*{\the\ht0}*{\the\ht2}%
903 \Expect*{\the\dp0}*{\the\dp2}%
904 \Expect*{\the\wd\boxB}*{\the\wd\boxA}%
905 \Expect*{\the\wd\boxC}*{\the\wd\boxA}%
906 \Expect*{\the\ht\boxB}*{\the\ht\boxA}%
907 \ifdim\ht\boxB=\ht\boxA
908 \else
909 \showbox\boxA
910 \showbox\boxB
911 \fi
912 \Expect*{\the\ht\boxC}*{\the\ht\boxA}%
913 \Expect*{\the\dp\boxB}*{\the\dp\boxA}%
914 \Expect*{\the\dp\boxC}*{\the\dp\boxA}%
915 \end{qstest}%
916 }

```

\SetTestVBox

```

917 \newcommand*\SetTestVBox}[2]{%
918 \setbox#1=\vbox{%
919 \hsize=0pt\relax
920 \hfuzz=100mm\relax
921 \expandafter\testdriver\expandafter{#2}%
922 }%
923 }

```

\TestInfo

```

924 \newcommand*\TestInfo}[2]{%
925 \par
926 \noindent
927 \copy#1\relax
928 \hfill
929 \begin{group}
930 \small
931 \ifdim\wd0=\wd2\relax
932 \ifdim\wd0=\wd4\relax
933 \else
934 \ifnum#1=2 %
935 \raisebox{.5\baselineskip}[Opt][Opt]{%
936 \llap{${}\$ }%

```

```

937         \rlap{width = \the\wd#1\relax\space$\{ $\}%
938         \phantom{width = 999.99999pt}}%
939         \qqquad
940     }%
941     \else
942         \ifnum#1=4 %
943             \rlap{width = \the\wd#1}}%
944             \phantom{width = 999.99999pt}}%
945             \qqquad
946         \fi
947     \fi
948     \fi
949     \else
950         \begingroup
951             \bfseries
952             \rlap{width = \the\wd#1}}%
953             \phantom{width = 999.99999pt}}%
954         \endgroup
955         \qqquad
956         \fi
957         \rlap{(#2)}}%
958         \phantom{(original soul)}}%
959     \endgroup
960     \par
961 }

```

#### \TestInfoVBox

```

962 \newcommand*{\TestInfoVBox}{%
963     \par
964     \noindent
965     \mbox{\vrule\relax\space\copy\boxA}%
966     \ifdim\ht\boxA=\ht\boxB
967     \else
968         \hfill
969         \begingroup
970             \small
971             \rlap{(soulutf8)}}%
972             \phantom{(soulutf8)}}%
973         \endgroup
974         \par
975         \noindent
976         \mbox{\vrule\relax\space\copy\boxB}%
977         \hfill
978         \begingroup
979             \small
980             \rlap{(soul)}}%
981             \phantom{(soulutf8)}}%
982         \endgroup
983     \fi
984     \par
985 }

```

#### \Test

```

986 \makeatletter
987 \newcommand*{\Test}[3]{%
988     \clearpage
989     \begingroup
990         \def\linebreak{\textbackslash linebreak }%
991         \def\{\textbackslash\textbackslash}%
992         \def\soulomit##1{%
993             \textbackslash soulomit\textbraceleft##1\textbraceright
994         }%

```

```

995 \def\mbox##1{%
996 \textbackslash mbox\textbraceleft##1\textbraceright
997 }%
998 \def\abc{\textbackslash abc }%
999 \def\texttt##1{%
1000 \textbackslash texttt\textbraceleft##1\textbraceright
1001 }%
1002 \def\c##1{%
1003 \textbackslash c\textbraceleft##1\textbraceright
1004 }%
1005  $\langle$ *utf8x)
1006 \count@=128 %
1007 \loop
1008 \ifnum\count@<256 %
1009 \lccode'\~=\count@
1010 \lowercase{%
1011 \def~{\noexpand~}%
1012 }%
1013 \advance\count@\@ne
1014 \repeat
1015  $\rangle$ /utf8x)
1016 \csname protected@edef\endcsname\x{\endgroup
1017 \noexpand\section{String #1: #2}%
1018 }%
1019 \x
1020 \TestDriver{so}{#1}{#2}{#3}%
1021 \TestDriver{caps}{#1}{#2}{#3}%
1022 \TestDriver{ul}{#1}{#2}{#3}%
1023 \TestDriver{st}{#1}{#2}{#3}%
1024 \TestDriver{hl}{#1}{#2}{#3}%
1025 \TestDriver{an}{#1}{#2}{#3}%
1026 \TestDriver{sy}{#1}{#2}{#3}%
1027 \TestDriver{soulH}{#1}{#2}{#3}%
1028 }
1029 \makeatother

```

### 3.2.3 Testing

```

1030 \begin{document}
1031 \Test A{}{}%
1032 \Test B{Hello World}{Hello World}%
1033 \Test C{%
1034 Umlauts: ^c3^84^c3^96^c3^9c^c3^a4^c3^b6^c3^bc^c3^9f%
1035 }{%
1036 Umlauts: \"A\"0\"U\"a\"o\"u\"ss
1037 }%
1038 \Test D{Euro: ^e2^82^ac}{Euro: \texteuro}%
1039 \Test E{Stra^c3^9fe}{Stra\ss e}%
1040 \Test F{R^c3^b6mer}{R\"omer}%
1041 \Test G{A^e2^80^93B^e2^80^94C}{A--B---C}%
1042 \Test H{X^c2^a0Y}{X~Y}%
1043 \Test I{A^c5^a4}{A\v T}%
1044 \Test J{A,B,,C,,,D,,,E}{A,B,,C,,,D,,,E}%
1045 \Test K{A'B''C''D''''E}{A'B''C''D''''E}%
1046 \Test L{A'B'C''D''''E}{A'B'C''D''''E}%
1047 \Test M{A$x$B}{A$x$B}%
1048 \Test N{A\mbox{abc}B}{A\mbox{abc}B}%
1049 \Test O{,TA''}{,TA''}%
1050 \Test P{{!'}Hola{?}}{!'}Hola{?}}%
1051 \Test Q{re-sent\slash x}{re-sent\slash x}%
1052 \Test R{new\\line}{new\\line}%
1053 \Test S{foo\linebreak bar}{foo\linebreak bar}%
1054 \Test T{Virchow{sche}}{Virchow{sche}}%

```

```

1055 \Test U{A\soulomit{foo}B}{A\soulomit{foo}B}%
1056 \Test V{A\texttt{xyz}B}{A\texttt{xyz}B}%
1057 \Test W{ffi}{ffi}%
1058 \Test X{f{}f{}i}{f{}f{}i}%
1059 \Test Y{f\mbox{}f\mbox{}i}{f\mbox{}f\mbox{}i}%
1060 \Test Z{\c{"a}}{\c{"a}}%
1061 <*test2 | test3>
1062 \let\abcx\UNDEFINED
1063 \def\abc{%
1064     \begingroup
1065         \def\abcx{x}%
1066         \abcx
1067     \endgroup
1068 }%
1069 \Test{AA}{[x\abc x]}{[xxx]}%
1070 </test2 | test3>
1071 \end{document}
1072 </test2 | test3 | test4 | test5>

```

## 4 Installation

### 4.1 Download

**Package.** This package is available on CTAN<sup>1</sup>:

[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 T<sub>E</sub>X Files” ([CTAN:tds/tds.pdf](#)). Directories with `texmf` in their name are usually organized this way.

### 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-T<sub>E</sub>X:

```
tex soulutf8.dtx
```

<sup>1</sup><http://ftp.ctan.org/tex-archive/>

**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 (`te $\TeX$` , `mik $\TeX$` , ...) relies on file name databases, you must refresh these. For example, `te $\TeX$`  users run `texhash` or `mktextlsr`.

#### 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 `pdf $\LaTeX$` :

```
pdflatex soulutf8.dtx
makeindex -s gind.ist soulutf8.idx
pdflatex soulutf8.dtx
makeindex -s gind.ist soulutf8.idx
pdflatex soulutf8.dtx
```

## 5 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](#).

## 6 History

[2007/09/09 v1.0]

- First version.

## 7 Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

Symbols	
<code>\"</code> .....	1036, 1040, 1060
<code>\#</code> .....	639, 701
<code>\%</code> .....	704
<code>\@</code> .....	640, 697
<code>\@PackageError</code> .....	131
<code>\@PackageInfo</code> .....	855, 860, 866, 873
<code>\@PackageWarningNoLine</code> .....	168
<code>\@ehc</code> .....	733
<code>\@empty</code> .....	861, 867, 874
<code>\@firstofone</code> .....	648, 651
<code>\@gobble</code> .....	487, 645, 653
<code>\@ifDefined</code> .....	729, 738, 745, 746
<code>\@ifdefinable</code> .....	737
<code>\@ifundefined</code> .....	730
<code>\@latex@error</code> .....	731
<code>\@ne</code> .....	198, 200, 202, 204, 283, 285, 287, 289, 893, 1013
<code>\@nil</code> .....	387, 390, 412, 416
<code>\@onelevel@sanitize</code> .....	<u>136</u> , 364, 438, 454
<code>\@undefined</code> .....	52, 112
<code>\[</code> .....	702
<code>\]</code> .....	225, 310, 698, 991, 1052
<code>\{</code> .....	637, 699, 937
<code>\}</code> .....	638, 700, 936
<code>\]</code> .....	703
<code>\~</code> .....	1009
<code>\_</code> .....	705, 778
A	
<code>\abc</code> .....	998, 1063, 1069
<code>\abcx</code> .....	1062, 1065, 1066
<code>\active</code> .....	105
<code>\advance</code> .....	198, 200, 202, 204, 283, 285, 287, 289, 678, 686, 1013
B	
<code>\afterassignment</code> .....	161
<code>\aftergroup</code> .....	26
<code>\an</code> .....	<u>776</u>
<code>\AtBeginDocument</code> .....	771
<code>\baselineskip</code> .....	935
<code>\begin</code> .....	859, 1030
<code>\bfseries</code> .....	951
<code>\body</code> .....	657, 661
<code>\boxA</code> .....	850, 864, 904, 905, 906, 907, 909, 912, 913, 914, 965, 966
<code>\boxB</code> .....	851, 871, 904, 906, 907, 910, 913, 966, 976
<code>\boxC</code> .....	852, 885, 905, 912, 914
<code>\break</code> .....	228, 248, 250, 313, 333, 335
C	
<code>\c</code> .....	1002, 1060
<code>\catcode</code> .....	3, 4, 5, 6, 7, 8, 9, 17, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 64, 65, 68, 69, 70, 71, 75, 76, 77, 78, 82, 84, 152, 153, 155, 156, 637, 638, 639, 640, 675, 684, 697, 698, 699, 700, 701, 702, 703, 704, 705, 706
<code>\cdot</code> .....	800
<code>\char</code> .....	778
<code>\chardef</code> .....	394, 421, 425, 429, 431
<code>\clearpage</code> .....	988
<code>\copy</code> .....	927, 965, 976
<code>\count@</code> .....	642, 671, 675, 677, 678, 682, 684, 685, 686, 1006, 1008, 1009, 1013
<code>\countdef</code> .....	642
<code>\csname</code> .....	10, 18, 44, 60, 67, 113, 136, 363, 364, 641,



	644, 647, 650, 689, 711, 723, 732, 739, 740, 747, 748, 856, 1016	\input . . . . . 124, 127, 135, 690	
\CurrentOption . . . . .	145	\iterate . . . . . 658, 660, 662	
<b>D</b>			
\DeclareOption . . . . .	116, 121, 145	<b>K</b>	
\DeclareRobustCommand . . . . .		\kern . . . . . 800	
. . . . . 114, 119, 776, 787, 812		<b>L</b>	
\DeclareUnicodeCharacter . . . . .	766	\lccode . . . . . 1009	
\discretionary . . . . .	796, 806	\leavevmode . . . . . 216, 301	
\documentclass . . . . .	112, 718	\lefthyphenmin . . . . . 790	
\dp . . . . .	903, 913, 914	\linebreak . . . . . 251, 253, 336, 338, 778, 780, 990, 1053	
<b>E</b>			
\empty . . . . .	13, 14	\llap . . . . . 936	
\end . . . . .	712, 915, 1071	\LoadCommand . . . . . 690, 707	
\endcsname . . . . . 10, 18, 44, 60, 67, 113, 136, 363, 364, 641, 644, 647, 650, 689, 711, 723, 732, 739, 740, 747, 748, 856, 1016		\LogTests . . . . . 849	
\endinginput . . . . .	26	\loop . . . . . 656, 672, 683, 1007	
\etex@unexpanded . . . . .		\lowercase . . . . . 1010	
. . . . . 447, 469, 477, 559, 584, 609, 628		<b>M</b>	
\Expect . . . . . 894, 895, 896, 902, 903, 904, 905, 906, 912, 913, 914		\makeatletter . . . . . 728, 775, 986	
<b>G</b>			
\gdef . . . . . 814, 842		\makeatother . . . . . 752, 1029	
<b>H</b>			
\hbox . . . . . 240, 325, 571, 583, 584, 596, 608, 609, 800, 805, 862, 869, 883		\maxdimen . . . . . 891, 892	
\hfil . . . . . 228, 313		\mbox 238, 323, 965, 976, 995, 1048, 1059	
\hfill . . . . . 928, 968, 977		\meaning . . . . . 139, 385, 409	
\hfuzz . . . . . 920		\MessageBreak . . . . . 169, 170, 171	
\hsize . . . . . 919		<b>N</b>	
\HstringA . . . . . 863, 894, 895		\NeedsTeXFormat . . . . . 717	
\HstringB . . . . . 870, 894		\newcommand . . . . . 115, 120, 729, 736, 744, 840, 853, 917, 924, 962, 987	
\HstringC . . . . . 884, 895		\newsavebox . . . . . 850, 851, 852	
\ht . . . . . 902, 906, 907, 912, 966		\next . . . . . 662, 664, 666	
<b>I</b>			
\if . . . . . 180, 217, 265, 302		\nobreak . . . . . 228, 233, 313, 318, 442	
\ifcase . . . . . 345, 348, 490, 498, 507, 515, 524, 532		\nobreakspace . . . . . 766	
\ifcat . . . . . 254, 256, 339, 341		\noindent . . . . . 926, 964, 975	
\ifdim . . . . . 897, 907, 931, 932, 966		\null . . . . . 213, 219, 227, 232, 298, 304, 312, 317, 441	
\ifetex@unexpanded . . . . . 151		<b>O</b>	
\ifnum . . . . . 184, 186, 188, 190, 269, 271, 273, 275, 677, 685, 934, 942, 1008		\origshow . . . . . 769, 772	
\ifSOUL@ignorespaces . . . . . 220, 305		<b>P</b>	
\ifx . . . . . 11, 14, 18, 44, 52, 55, 112, 113, 136, 165, 193, 194, 195, 196, 197, 199, 201, 203, 210, 215, 225, 230, 234, 238, 240, 242, 248, 251, 278, 279, 280, 281, 282, 284, 286, 288, 295, 300, 310, 315, 319, 323, 325, 327, 333, 336, 395, 397, 399, 420, 424, 428, 439, 455, 461, 641, 644, 647, 650, 689		\PackageError . . . . . 117, 122, 131	
\immediate . . . . . 20, 46		\PackageInfo . . . . . 23	
\IncludeTests . . . . . 848		\par . . . . . 215, 216, 300, 301, 925, 960, 963, 974, 984	
<b>J</b>			
<b>K</b>			
<b>L</b>			
<b>M</b>			
<b>N</b>			
<b>O</b>			
<b>P</b>			
<b>Q</b>			
<b>R</b>			
\PackageError . . . . . 117, 122, 131		\PackageInfo . . . . . 23	
\PackageInfo . . . . . 23		\par . . . . . 215, 216, 300, 301, 925, 960, 963, 974, 984	
\par . . . . . 215, 216, 300, 301, 925, 960, 963, 974, 984		\PassOptionsToPackage . . . . . 145	
\PassOptionsToPackage . . . . . 145		\phantom . . . . . 938, 944, 953, 958, 972, 981	
\phantom . . . . . 938, 944, 953, 958, 972, 981		\ProcessOptions . . . . . 146	
\ProcessOptions . . . . . 146		\protect . . . . . 542, 554, 567, 579, 592, 604	
\protect . . . . . 542, 554, 567, 579, 592, 604		\ProvidesPackage . . . . . 15, 61	
\ProvidesPackage . . . . . 15, 61		<b>Q</b>	
<b>Q</b>			
\qqquad . . . . . 939, 945, 955		<b>R</b>	
<b>R</b>			
\raisebox . . . . . 935		\RangeCatcodeInvalid . . . . . . . . . . 681, 693, 694, 695, 696	
\RangeCatcodeInvalid . . . . . . . . . . 681, 693, 694, 695, 696		\repeat . . . . . 656, 668, 679, 687, 1014	
\repeat . . . . . 656, 668, 679, 687, 1014		\RequirePackage . . . . . 147, 148, 149	
\RequirePackage . . . . . 147, 148, 149		\RestoreCatcodes . . . . . 670, 673, 674, 708	
\RestoreCatcodes . . . . . 670, 673, 674, 708		\RestoreOrg . . . . . 744, 868, 875, 876, 877, 878, 879, 880, 881, 882	
\RestoreOrg . . . . . 744, 868, 875, 876, 877, 878, 879, 880, 881, 882			

<code>\righthyphenmin</code>	791	<code>\SOUL@everylowerthan</code>	784, 835
<code>\rlap</code>	937, 943, 952, 957, 971, 980	<code>\SOUL@everyspace</code>	222, 228, 233, 253, 307, 313, 318, 338, 442, 778, 793, 817
<b>S</b>			
<code>\SavedUnexpanded</code>	715, 763	<code>\SOUL@everysyllable</code>	782, 794, 829
<code>\SaveOrg</code>	736, 753, 754, 755, 756, 757, 758, 759, 760, 761	<code>\SOUL@everytoken</code>	783, 832
<code>\section</code>	1017	<code>\SOUL@exhyphen</code>	237, 322, 458, 464
<code>\setbox</code>	862, 869, 883, 918	<code>\SOUL@flushapo</code>	189, 195, 208, 274, 280, 293, 506
<code>\SetTestVBox</code>	864, 871, 885, 917	<code>\SOUL@flushcomma</code>	187, 194, 207, 272, 279, 292, 489
<code>\show</code>	769, 772	<code>\SOUL@flushgrave</code>	191, 196, 209, 276, 281, 294, 523
<code>\showbox</code>	899, 900, 909, 910	<code>\SOUL@flushminus</code>	185, 193, 206, 270, 278, 291
<code>\showboxbreadth</code>	892	<code>\SOUL@grave</code>	190, 204, 275, 289, 524, 530, 532, 538
<code>\showboxdepth</code>	891	<code>\SOUL@hyphkern</code>	797, 804
<code>\slash</code>	234, 319, 1051	<code>\SOUL@ignorespacesfalse</code>	182, 267
<code>\small</code>	930, 970, 979	<code>\SOUL@ignorespacestrue</code>	229, 314
<code>\SOUL@</code>	785, 810, 838	<code>\SOUL@minus</code>	184, 198, 269, 283
<code>\SOUL@@</code>	180, 193, 194, 195, 196, 197, 199, 201, 203, 210, 215, 217, 225, 230, 234, 238, 240, 242, 248, 251, 254, 256, 265, 278, 279, 280, 281, 282, 284, 286, 288, 295, 300, 302, 310, 315, 319, 323, 325, 327, 333, 336, 339, 341, 385, 409	<code>\SOUL@n</code>	179, 211, 216, 224, 239, 241, 243, 255, 257, 259, 262, 264, 296, 301, 309, 324, 326, 328, 340, 342, 349, 352, 354, 356, 360
<code>\SOuL@addfouroctets</code>	356, 475	<code>\SOuL@next</code>	443, 445, 450, 459, 465, 467, 473
<code>\SOUL@addgroup</code>	255, 340, 540	<code>\SOuL@octets</code>	345, 348, 394, 421, 425, 429, 431
<code>\SOUL@addmath</code>	257, 342, 565	<code>\SOuL@orgDeclareOption</code>	116, 121
<code>\SOUL@addprotect</code>	239, 241, 324, 326, 590	<code>\SOuL@orgDeclareRobustCommand</code>	114, 119
<code>\SOuL@addthreecoctets</code>	354, 452	<code>\SOuL@orgnewcommand</code>	115, 120
<code>\SOUL@addtoken</code>	446, 447, 468, 469, 476, 477, 615	<code>\SOuL@orgPackageError</code>	117, 122
<code>\SOuL@addtwooctets</code>	352, 436	<code>\SOuL@orgSOUL@error</code>	128, 132
<code>\SOuL@analyzeutfviii</code>	344, 381	<code>\SOUL@postamble</code>	779, 820
<code>\SOuL@analyzeutfviiix</code>	346, 405	<code>\SOUL@preamble</code>	777, 789, 813
<code>\SOUL@apo</code>	188, 202, 273, 287, 507, 513, 515, 521	<code>\SOuL@redefine</code>	158, 178, 479, 489, 506, 523, 540, 565, 590, 615
<code>\SOuL@AtEnd</code>	80, 81, 634	<code>\SOuL@remainingtoken</code>	484, 486
<code>\SOuL@charhash</code>	377, 410, 415	<code>\SOuL@restorelatexcmds</code>	118, 125
<code>\SOUL@charkern</code>	807	<code>\SOUL@scan</code>	179, 216, 224, 246, 264, 301, 309, 331, 443, 459, 465, 551, 563, 576, 588, 601, 613, 623, 632
<code>\SOuL@charthree</code>	379, 424	<code>\SOUL@sethyphenchar</code>	798
<code>\SOuL@chartwo</code>	378, 428	<code>\SOUL@setkern</code>	797, 804, 807
<code>\SOuL@checkutfviii</code>	384, 389	<code>\SOUL@setup</code>	788
<code>\SOuL@checkutfviiix</code>	408, 414	<code>\SOUL@spaceskip</code>	245, 330
<code>\SOuL@cmd</code>	160, 165, 167, 176	<code>\SOUL@spc</code>	180, 217, 265, 302
<code>\SOuL@cmdcheck</code>	161, 164	<code>\SOUL@splittoken</code>	479
<code>\SOUL@comma</code>	186, 200, 271, 285, 490, 496, 498, 504	<code>\SOUL@stop</code>	210, 295, 479, 486
<code>\SOuL@defsanitizedstring</code>	362, 366, 367, 368, 369, 370, 371, 372, 373	<code>\SOuL@stringcsnameu</code>	411, 416
<code>\SOUL@dotoken</code>	259, 349	<code>\SOuL@stringemdash</code>	375, 461
<code>\SOUL@doword</code>	212, 218, 226, 231, 235, 244, 249, 252, 297, 303, 311, 316, 320, 329, 334, 337, 440, 456, 462	<code>\SOuL@stringendash</code>	374, 455
<code>\SOuL@empty</code>	380, 420	<code>\SOuL@stringfour</code>	399
<code>\SOUL@error</code>	127	<code>\SOuL@stringmacrocolon</code>	410, 415
<code>\SOUL@eval</code>	178	<code>\SOuL@stringnobreakspace</code>	376, 439
<code>\SOUL@eventuallyexhyphen</code>	213, 219, 227, 232, 236, 298, 304, 312, 317, 321, 441, 457, 463	<code>\SOuL@stringoctets</code>	386, 390
<code>\SOUL@everyexhyphen</code>	781, 803, 826	<code>\SOuL@stringthree</code>	397
<code>\SOUL@everyhyphen</code>	780, 795, 823	<code>\SOuL@stringtwo</code>	395
		<code>\SOuL@stringundeferr</code>	411, 416

<code>\SOuL@stringUTFviii</code>	386, 390	<code>\texttt</code>	778, 999, 1056
<code>\SOUL@syllable</code>	794	<code>\the</code>	68, 69, 70, 71, 82, 492, 494, 500, 502, 509, 511, 517, 519, 526, 528, 534, 536, 545, 557, 570, 582, 595, 607, 618, 627, 675, 783, 794, 896, 902, 903, 904, 905, 906, 912, 913, 914, 937, 943, 952
<code>\SOuL@temp</code>	162, 165, 167, 169, 172, 393, 395, 397, 399, 419, 420, 423, 424, 427, 428, 437, 438, 439, 453, 454, 455, 461	<code>\thr@</code>	184, 269, 398, 429
<code>\SOUL@token</code>	480, 483, 783	<code>\TMP@EnsureCode</code>	79, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111
<code>\SOUL@word</code>	481, 487, 492, 494, 500, 502, 509, 511, 517, 519, 526, 528, 534, 536, 544, 545, 556, 557, 569, 570, 581, 582, 594, 595, 606, 607, 617, 618, 626, 627	<code>\tracingonline</code>	893
<code>\SOULH</code>	814	<code>\tw@</code>	186, 188, 190, 271, 273, 275, 396, 431
<code>\soulH</code>	812		
<code>\soulHadd</code>	815, 818, 821, 824, 827, 830, 833, 836, 840	<b>U</b>	
<code>\soulHstring</code>	842, 843, 861, 863, 867, 870, 874, 884	<code>\UNDEFINED</code>	716, 1062
<code>\soulomit</code>	242, 243, 327, 328, 992, 1055	<code>\unexpanded</code>	715, 716, 763
<code>\space</code>	793, 937, 965, 976	<code>\unskip</code>	228, 313
<code>\spaceskip</code>	245, 330	<code>\usepackage</code>	719, 720, 721, 722, 724, 725, 726, 727, 762, 765, 770, 847
<code>\ss</code>	1036, 1039	<b>V</b>	
<code>\strip@prefix</code>	139, 142	<code>\v</code>	1043
<code>\subsection</code>	854	<code>\vbox</code>	918
<code>\sy</code>	787	<code>\vrule</code>	965, 976
<b>T</b>		<b>W</b>	
<code>\Test</code>	692, 710, 986, 1031, 1032, 1033, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1069	<code>\wd</code>	896, 897, 904, 905, 931, 932, 937, 943, 952
<code>\TestDriver</code>	853, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027	<code>\write</code>	20, 46
<code>\testdriver</code>	856, 862, 869, 883, 921	<b>X</b>	
<code>\TestInfo</code>	887, 888, 889, 924	<code>\x</code>	10, 11, 14, 19, 23, 25, 45, 50, 60, 66, 74, 382, 392, 406, 418, 492, 494, 500, 502, 509, 511, 517, 519, 526, 528, 534, 536, 543, 549, 555, 562, 568, 574, 580, 587, 593, 599, 605, 612, 616, 622, 625, 631, 1016, 1019
<code>\TestInfoVBox</code>	890, 962	<b>Z</b>	
<code>\teststringA</code>	857, 862, 864	<code>\z@</code>	402, 421, 496, 504, 513, 521, 530, 538, 790, 791
<code>\teststringB</code>	858, 869, 871, 883, 885		
<code>\textbackslash</code>	990, 991, 993, 996, 998, 1000, 1003		
<code>\textbraceleft</code>	993, 996, 1000, 1003		
<code>\textbraceright</code>	993, 996, 1000, 1003		
<code>\texteuro</code>	1038		