

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

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 \TeX	5
2.3.2	\LaTeX	6
2.3.3	$\varepsilon\text{-TeX}$	6
2.4	Macro for redefinitions	6
2.5	Redefinition of <code>\SOUL@eval</code>	7
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	13
3	Test	16
3.1	Catcode checks for loading	16
3.2	\LaTeX tests	17
3.2.1	Driver definitions	18
3.2.2	Test macros	20
3.2.3	Testing	23
4	Installation	24
4.1	Download	24
4.2	Bundle installation	24
4.3	Package installation	25
4.4	Refresh file name databases	25
4.5	Some details for the interested	25

5	Catalogue	26
6	References	26
7	History	27
	[2007/09/09 v1.0]	27
8	Index	27

1 Documentation

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

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

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

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

1.1 Patch

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

- A problem with additional levels of curly braces is fixed. As advantage more 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}%
68 [2007/09/09 v1.0 Permit use of UTF-8 characters in soul (HO)]%

```

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_TX.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}[2007/09/09]%
166 \RequirePackage{etexcmds}[2007/09/09]%
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@ignorespacestrue
```

[illegible]

Now the redefined version follows.

```

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\%
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\%
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\%
330 \SOUL@everySPACE{\unskip\nobreak\hfil\break}%
331 \SOUL@ignorespacestrue
332 \else\ifx\SOUL@@~%
333 \SOUL@doword
334 \SOUL@eventuallyexhyphen\%
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

```

[illegible]

```

379 \def\S0uL@defsanitizedstring#1#2{%
380   \expandafter\def\csname S0uL@string#1\endcsname{#2}%
381   \expandafter\@onelevel@sanitize\csname S0uL@string#1\endcsname
382 }
383 \S0uL@defsanitizedstring{UTFviii}{UTFviii@}
384 \S0uL@defsanitizedstring{octets}{@octets}
385 \S0uL@defsanitizedstring{two}{two}
386 \S0uL@defsanitizedstring{three}{three}
387 \S0uL@defsanitizedstring{four}{four}
388 \S0uL@defsanitizedstring{macrocolon}{macro:}
389 \S0uL@defsanitizedstring{csnameu}{\csname u8-}
390 \S0uL@defsanitizedstring{undeferr}{utf@viii@undeferr}
391 \def\S0uL@stringendash{^^e2^^80^^93}
392 \def\S0uL@stringemdash{^^e2^^80^^94}
393 \def\S0uL@stringnobreakspace{^^c2^^a0}
394 \edef\S0uL@charhash{\string #}
395 \edef\S0uL@chartwo{\string 2}
396 \edef\S0uL@charthree{\string 3}
397 \def\S0uL@empty{}

```

```

410 \def\SouL@temp{#2}%
411 \chardef\SouL@octets=%
412 \ifx\SouL@temp\SouL@stringtwo
413 \tw@
414 \else\ifx\SouL@temp\SouL@stringthree
415 \thr@@
416 \else\ifx\SouL@temp\SouL@stringfour
417 4 %
418 \else
419 \z@
420 \fi\fi\fi
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 Test

3.1 Catcode checks for loading

```
653 (*test1)
654 \catcode'\{=1 %
655 \catcode'\}=2 %
656 \catcode'\#=6 %
657 \catcode'\@=11 %
658 \expandafter\ifx\csname count@\endcsname\relax
659 \countdef\count@=255 %
660 \fi
661 \expandafter\ifx\csname @gobble\endcsname\relax
662 \long\def\@gobble#1{}%
663 \fi
664 \expandafter\ifx\csname @firstofone\endcsname\relax
665 \long\def\@firstofone#1{#1}%
666 \fi
667 \expandafter\ifx\csname loop\endcsname\relax
668 \expandafter\@firstofone
669 \else
670 \expandafter\@gobble
671 \fi
672 {%
673 \def\loop#1\repeat{%
674 \def\body{#1}%
675 \iterate
676 }%
677 \def\iterate{%
678 \body
679 \let\next\iterate
680 \else
681 \let\next\relax
682 \fi
683 \next
684 }%
685 \let\repeat=\fi
686 }%
687 \def\RestoreCatcodes{}
688 \count@=0 %
689 \loop
690 \edef\RestoreCatcodes{%
691 \RestoreCatcodes
692 \catcode\the\count@=\the\catcode\count@\relax
693 }%
694 \ifnum\count@<255 %
695 \advance\count@ 1 %
696 \repeat
697
698 \def\RangeCatcodeInvalid#1#2{%
699 \count@=#1\relax
700 \loop
701 \catcode\count@=15 %
702 \ifnum\count@<#2\relax
703 \advance\count@ 1 %
704 \repeat
705 }
706 \def\RangeCatcodeCheck#1#2#3{%
```



```

707 \count@=#1\relax
708 \loop
709 \ifnum#3=\catcode\count@
710 \else
711 \errmessage{%
712 Character \the\count@\space
713 with wrong catcode \the\catcode\count@\space
714 instead of \number#3%
715 }%
716 \fi
717 \ifnum\count@<#2\relax
718 \advance\count@ 1 %
719 \repeat
720 }
721 \def\space{ }
722 \expandafter\ifx\csname LoadCommand\endcsname\relax
723 \def\LoadCommand{\input soulutf8.sty\relax}%
724 \fi
725 \def\Test{%
726 \RangeCatcodeInvalid{0}{47}%
727 \RangeCatcodeInvalid{58}{64}%
728 \RangeCatcodeInvalid{91}{96}%
729 \RangeCatcodeInvalid{123}{255}%
730 \catcode'\@=12 %
731 \catcode'\=0 %
732 \catcode'\%=14 %
733 \LoadCommand
734 \RangeCatcodeCheck{0}{36}{15}%
735 \RangeCatcodeCheck{37}{37}{14}%
736 \RangeCatcodeCheck{38}{47}{15}%
737 \RangeCatcodeCheck{48}{57}{12}%
738 \RangeCatcodeCheck{58}{63}{15}%
739 \RangeCatcodeCheck{64}{64}{12}%
740 \RangeCatcodeCheck{65}{90}{11}%
741 \RangeCatcodeCheck{91}{91}{15}%
742 \RangeCatcodeCheck{92}{92}{0}%
743 \RangeCatcodeCheck{93}{96}{15}%
744 \RangeCatcodeCheck{97}{122}{11}%
745 \RangeCatcodeCheck{123}{255}{15}%
746 \RestoreCatcodes
747 }
748 \Test
749 \csname @@end\endcsname
750 \end
751 </test1>

```

3.2 L^AT_EX tests

```

752 <*test2 | test3 | test4 | test5>
753 <noetex>\let\SavedUnexpanded\unexpanded
754 <noetex>\let\unexpanded\UNDEFINED
755 \NeedsTeXFormat{LaTeX2e}
756 \documentclass{article}
757 \usepackage[T1]{fontenc}
758 \usepackage{lmodern}
759 \usepackage{textcomp}
760 \usepackage[ngerman]{babel}
761 \expandafter\def\csname ver@hypdestopt.sty\endcsname{

```

```

762 \usepackage[unicode]{hyperref}
763 \usepackage{bookmark}
764 \usepackage{color}
765 \usepackage{soul}[2003/11/17]
766 \makeatletter
767 \newcommand*{\@ifDefined}[1]{%
768   \@ifundefined{#1}{%
769     \@latex@error{%
770       \expandafter\let\csname#1\endcsname undefined%
771     }\@ehc
772   }%
773 }
774 \newcommand*{\SaveOrg}[1]{%
775   \@ifdefinable{ORG@SOUL@#1}{%
776     \@ifDefined{SOUL@#1}{%
777       \expandafter\let\csname ORG@SOUL@#1\expandafter\endcsname
778         \csname SOUL@#1\endcsname
779     }%
780   }%
781 }
782 \newcommand*{\RestoreOrg}[1]{%
783   \@ifDefined{ORG@SOUL@#1}{%
784     \@ifDefined{SOUL@#1}{%
785       \expandafter\let\csname SOUL@#1\expandafter\endcsname
786         \csname ORG@SOUL@#1\endcsname
787     }%
788   }%
789 }
790 \makeatother
791 \SaveOrg{eval}
792 \SaveOrg{splittoken}
793 \SaveOrg{flushcomma}
794 \SaveOrg{flushapo}
795 \SaveOrg{grave}
796 \SaveOrg{addgroup}
797 \SaveOrg{addmath}
798 \SaveOrg{addprotect}
799 \SaveOrg{addtoken}
800 \usepackage{soulutf8}[2007/09/09]
801 \noetex\let\unexpanded\SavedUnexpanded

```

Input encoding utf8.

```

802 (*utf8)
803 \usepackage[utf8]{inputenc}
804 \DeclareUnicodeCharacter{00A0}{\nobreakspace}
805 /utf8)

```

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

```

806 (*utf8x)
807 \let\origshow\show
808 \usepackage[utf8x]{inputenc}
809 \AtBeginDocument{%
810   \let\show\origshow
811 }
812 /utf8x)

```

3.2.1 Driver definitions

```

813 \makeatletter

```

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

```

814 \DeclareRobustCommand*\an{%
815   \def\SOUL@preamble{$^{^P}$}%
816   \def\SOUL@everyspace##1{##1\texttt{\char'\ }\linebreak[2]}%
817   \def\SOUL@postamble{$^{^E}$}%
818   \def\SOUL@everyhyphen{$^{^-}$\linebreak[1]}%
819   \def\SOUL@everyexhyphen##1{##1$^{^=}$}%
820   \def\SOUL@everysyllable{$^{^S}$}%
821   \def\SOUL@everytoken{\the\SOUL@token$^{^T}$}%
822   \def\SOUL@everylowerthan{$^{^L}$}%
823   \SOUL@
824 }

```

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

```

825 \DeclareRobustCommand*\sy{%
826   \SOUL@setup
827   \def\SOUL@preamble{%
828     \lefthyphenmin=\z@
829     \righthyphenmin=\z@
830   }%
831   \def\SOUL@everyspace##1{##1\space}%
832   \def\SOUL@everysyllable{\the\SOUL@syllable}%
833   \def\SOUL@everyhyphen{%
834     \discretionary{%
835       \SOUL@setkern\SOUL@hyphkern
836       \SOUL@sethyphenchar
837     }{}{%
838       \hbox{\kern1pt$\cdot$}%
839     }%
840   }%
841   \def\SOUL@everyexhyphen##1{%
842     \SOUL@setkern\SOUL@hyphkern
843     \hbox{##1}%
844     \discretionary{}{}{%
845       \SOUL@setkern\SOUL@charkern
846     }%
847   }%
848   \SOUL@
849 }

```

`\soulH`

```

850 \DeclareRobustCommand*\soulH{%
851   \def\SOUL@preamble{%
852     \gdef\SOULH{}%
853     \soulHadd{ }%
854   }%
855   \def\SOUL@everyspace##1{%
856     \soulHadd{Sp}%
857   }%
858   \def\SOUL@postamble{%
859     \soulHadd{ }%
860   }%
861   \def\SOUL@everyhyphen{%
862     \soulHadd{Hy}%
863   }%
864   \def\SOUL@everyexhyphen##1{%
865     \soulHadd{Exhy}%
866   }%

```

```

867 \def\SOUL@everysyllable{%
868   \soulHadd{Sy}%
869 }%
870 \def\SOUL@everytoken{%
871   \soulHadd{T}%
872 }%
873 \def\SOUL@everylowerthan{%
874   \soulHadd{<}%
875 }%
876 \SOUL@
877 }
878 \newcommand*{\soulHadd}[1]{%
879   #1%
880   \expandafter\gdef\expandafter\soulHstring\expandafter{%
881     \soulHstring
882     #1%
883   }%
884 }

```

3.2.2 Test macros

```

885 \usepackage{qstest}
886 \IncludeTests{*}
887 \LogTests{log}{*}{*}

888 \newsavebox\boxA
889 \newsavebox\boxB
890 \newsavebox\boxC

\TestDriver

891 \newcommand*{\TestDriver}[4]{%
892   \subsection{Driver #1}%
893   \@PackageInfo{soulutf8-test}{* Driver #1}%
894   \expandafter\let\expandafter\testdriver\csname #1\endcsname
895   \def\teststringA{#3}%
896   \def\teststringB{#4}%
897   \begin{qstest}{string-#2-driver-#1}{string,#2,driver,#1}%
898     \@PackageInfo{soulutf8-test}{* Version: soulutf8}%
899     \global\let\soulHstring\@empty
900     \setbox0=\hbox{\expandafter\testdriver\expandafter{\teststringA}}%
901     \let\HstringA\soulHstring
902     \SetTestVBox{\boxA}{\teststringA}%
903     %
904     \@PackageInfo{soulutf8-test}{* Version: soul + patch}%
905     \global\let\soulHstring\@empty
906     \RestoreOrg{eval}%
907     \setbox2=\hbox{\expandafter\testdriver\expandafter{\teststringB}}%
908     \let\HstringB\soulHstring
909     \SetTestVBox{\boxB}{\teststringB}%
910     %
911     \@PackageInfo{soulutf8-test}{* Version: original soul}%
912     \global\let\soulHstring\@empty
913     \RestoreOrg{splittoken}%
914     \RestoreOrg{flushcomma}%
915     \RestoreOrg{flushapo}%
916     \RestoreOrg{grave}%
917     \RestoreOrg{addgroup}%
918     \RestoreOrg{addmath}%
919     \RestoreOrg{addprotect}%

```

```

920 \RestoreOrg{addtoken}%
921 \setbox4=\hbox{\expandafter\testdriver\expandafter{\teststringB}}%
922 \let\HstringC\soulHstring
923 \SetTestVBox{\boxC}{\teststringB}%
924 %
925 \TestInfo{0}{soulutf8}%
926 \TestInfo{2}{soul + patch}%
927 \TestInfo{4}{original soul}%
928 \TestInfoVBox
929 \showboxdepth=\maxdimen
930 \showboxbreadth=\maxdimen
931 \tracingonline=\@ne
932 \Expect*{\HstringA}*{\HstringB}%
933 \Expect*{\HstringA}*{\HstringC}%
934 \Expect*{\the\wd0}*{\the\wd2}%
935 \ifdim\wd0=\wd2 %
936 \else
937 \showbox0 %
938 \showbox2 %
939 \fi
940 \Expect*{\the\ht0}*{\the\ht2}%
941 \Expect*{\the\dp0}*{\the\dp2}%
942 \Expect*{\the\wd\boxB}*{\the\wd\boxA}%
943 \Expect*{\the\wd\boxC}*{\the\wd\boxA}%
944 \Expect*{\the\ht\boxB}*{\the\ht\boxA}%
945 \ifdim\ht\boxB=\ht\boxA
946 \else
947 \showbox\boxA
948 \showbox\boxB
949 \fi
950 \Expect*{\the\ht\boxC}*{\the\ht\boxA}%
951 \Expect*{\the\dp\boxB}*{\the\dp\boxA}%
952 \Expect*{\the\dp\boxC}*{\the\dp\boxA}%
953 \end{qstest}%
954 }

```

\SetTestVBox

```

955 \newcommand*\SetTestVBox}[2]{%
956 \setbox#1=\vbox{%
957 \hsize=0pt\relax
958 \hfuzz=100mm\relax
959 \expandafter\testdriver\expandafter{#2}%
960 }%
961 }

```

\TestInfo

```

962 \newcommand*\TestInfo}[2]{%
963 \par
964 \noindent
965 \copy#1\relax
966 \hfill
967 \begin{group}
968 \small
969 \ifdim\wd0=\wd2\relax
970 \ifdim\wd0=\wd4\relax
971 \else
972 \ifnum#1=2 %
973 \raisebox{.5\baselineskip}[0pt][0pt]{%

```

```

974         \llap{$\}$ }%
975         \rlap{width = \the\wd#1\relax\space$\{$}%
976         \phantom{width = 999.99999pt}%
977         \qqquad
978     }%
979     \else
980         \ifnum#1=4 %
981             \rlap{width = \the\wd#1}%
982             \phantom{width = 999.99999pt}%
983             \qqquad
984         \fi
985     \fi
986 \fi
987 \else
988     \begingroup
989     \bfseries
990     \rlap{width = \the\wd#1}%
991     \phantom{width = 999.99999pt}%
992 \endgroup
993 \qqquad
994 \fi
995 \rlap{(#2)}%
996 \phantom{(original soul)}%
997 \endgroup
998 \par
999 }

```

\TestInfoVBox

```

1000 \newcommand*{\TestInfoVBox}{%
1001     \par
1002     \noindent
1003     \mbox{\vrule\relax\space\copy\boxA}%
1004     \ifdim\ht\boxA=\ht\boxB
1005     \else
1006         \hfill
1007         \begingroup
1008             \small
1009             \rlap{(soulutf8)}%
1010             \phantom{(soulutf8)}%
1011         \endgroup
1012     \par
1013     \noindent
1014     \mbox{\vrule\relax\space\copy\boxB}%
1015     \hfill
1016     \begingroup
1017         \small
1018         \rlap{(soul)}%
1019         \phantom{(soulutf8)}%
1020     \endgroup
1021 \fi
1022 \par
1023 }

```

\Test

```

1024 \makeatletter
1025 \newcommand*{\Test}[3]{%
1026     \clearpage
1027     \begingroup

```

```

1028 \def\linebreak{\textbackslash linebreak }%
1029 \def\\{\textbackslash\textbackslash}%
1030 \def\soulomit##1{%
1031 \textbackslash soulomit\textbraceleft##1\textbraceright
1032 }%
1033 \def\mbox##1{%
1034 \textbackslash mbox\textbraceleft##1\textbraceright
1035 }%
1036 \def\abc{\textbackslash abc }%
1037 \def\texttt##1{%
1038 \textbackslash texttt\textbraceleft##1\textbraceright
1039 }%
1040 \def\c##1{%
1041 \textbackslash c\textbraceleft##1\textbraceright
1042 }%
1043 (*utf8x)
1044 \count@=128 %
1045 \loop
1046 \ifnum\count@<256 %
1047 \lccode'\~=\count@
1048 \lowercase{%
1049 \def~{\noexpand~}%
1050 }%
1051 \advance\count@\@ne
1052 \repeat
1053 (/utf8x)
1054 \csname protected@edef\endcsname\x{\endgroup
1055 \noexpand\section{String #1: #2}%
1056 }%
1057 \x
1058 \TestDriver{so}{#1}{#2}{#3}%
1059 \TestDriver{caps}{#1}{#2}{#3}%
1060 \TestDriver{ul}{#1}{#2}{#3}%
1061 \TestDriver{st}{#1}{#2}{#3}%
1062 \TestDriver{hl}{#1}{#2}{#3}%
1063 \TestDriver{an}{#1}{#2}{#3}%
1064 \TestDriver{sy}{#1}{#2}{#3}%
1065 \TestDriver{soulH}{#1}{#2}{#3}%
1066 }
1067 \makeatother

```

3.2.3 Testing

```

1068 \begin{document}
1069 \Test A{}{}%
1070 \Test B{Hello World}{Hello World}%
1071 \Test C{%
1072 Umlauts: ^^c3^^84^^c3^^96^^c3^^9c^^c3^^a4^^c3^^b6^^c3^^bc^^c3^^9f%
1073 }{%
1074 Umlauts: \"A\"O\"U\"a\"o\"u\"ss
1075 }%
1076 \Test D{Euro: ^^e2^^82^^ac}{Euro: \texteuro}%
1077 \Test E{Stra^^c3^^9fe}{Stra\ss e}%
1078 \Test F{R^^c3^^b6mer}{R\"omer}%
1079 \Test G{A^^e2^^80^^93B^^e2^^80^^94C}{A--B---C}%
1080 \Test H{X^^c2^^a0Y}{X~Y}%
1081 \Test I{A^^c5^^a4}{A\v T}%
1082 \Test J{A,B,,C,,,D,,,E}{A,B,,C,,,D,,,E}%

```

```

1083 \Test K{A'B'C''D''E}{A'B'C''D''E}%
1084 \Test L{A'B'C''D''E}{A'B'C''D''E}%
1085 \Test M{A$x$B}{A$x$B}%
1086 \Test N{A\mbox{abc}B}{A\mbox{abc}B}%
1087 \Test O{,TA''}{,TA''}%
1088 \Test P{{!'}Hola{?'}}{{!'}Hola{?'}}%
1089 \Test Q{re-sent\slash x}{re-sent\slash x}%
1090 \Test R{new\\line}{new\\line}%
1091 \Test S{foo\linebreak bar}{foo\linebreak bar}%
1092 \Test T{Virchow{sche}}{Virchow{sche}}%
1093 \Test U{A\soulomit{foo}B}{A\soulomit{foo}B}%
1094 \Test V{A\texttt{xyz}B}{A\texttt{xyz}B}%
1095 \Test W{ffi}{ffi}%
1096 \Test X{f{}f{}i}{f{}f{}i}%
1097 \Test Y{f\mbox{ }f\mbox{ }i}{f\mbox{ }f\mbox{ }i}%
1098 \Test Z{\c{"a}}{\c{"a}}%
1099 (*test2 | test3)
1100 \let\abcx\UNDEFINED
1101 \def\abc{%
1102     \begin{group}
1103         \def\abcx{x}%
1104         \abcx
1105     \end{group}
1106 }%
1107 \Test{AA}{[[x\abc x]]}[[xxx]]%
1108 /test2 | test3)
1109 \end{document}
1110 /test2 | test3 | test4 | test5)

```

4 Installation

4.1 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.

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: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
```

¹<http://ftp.ctan.org/tex-archive/>

Script installation. Check the directory `TDS:scripts/oberdiek/` for scripts that need further installation steps. Package `attachfile2` comes with the Perl script `pdfatfi.pl` that should be installed in such a way that it can be called as `pdfatfi`. Example (linux):

```
chmod +x scripts/oberdiek/pdfatfi.pl
cp scripts/oberdiek/pdfatfi.pl /usr/local/bin/
```

4.3 Package installation

Unpacking. The `.dtx` file is a self-extracting `docstrip` archive. The files are extracted by running the `.dtx` through plain \TeX :

```
tex soulutf8.dtx
```

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

```
soulutf8.sty      → tex/generic/oberdiek/soulutf8.sty
soulutf8.pdf      → doc/latex/oberdiek/soulutf8.pdf
test/soulutf8-test1.tex → doc/latex/oberdiek/test/soulutf8-test1.tex
test/soulutf8-test2.tex → doc/latex/oberdiek/test/soulutf8-test2.tex
test/soulutf8-test3.tex → doc/latex/oberdiek/test/soulutf8-test3.tex
test/soulutf8-test4.tex → doc/latex/oberdiek/test/soulutf8-test4.tex
test/soulutf8-test5.tex → doc/latex/oberdiek/test/soulutf8-test5.tex
soulutf8.dtx      → source/latex/oberdiek/soulutf8.dtx
```

If you have a `docstrip.cfg` that configures and enables `docstrip`'s TDS installing feature, then some files can already be in the right place, see the documentation of `docstrip`.

4.4 Refresh file name databases

If your \TeX distribution (`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 pdfL^AT_EX:

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

5 Catalogue

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

```
1111 (*catalogue)
1112 <?xml version='1.0' encoding='us-ascii'?>
1113 <!DOCTYPE entry SYSTEM 'catalogue.dtd'>
1114 <entry datestamp='$Date$' modifier='$Author$' id='soulutf8'>
1115   <name>soulutf8</name>
1116   <caption>Permit use of UTF-8 characters in soul.</caption>
1117   <authorref id='auth:oberdiek'/>
1118   <copyright owner='Heiko Oberdiek' year='2007' />
1119   <license type='lppl1.3' />
1120   <version number='1.0' />
1121   <description>
1122     This package extends package <xref refid='soul'>soul</xref> and adds
1123     some support for UTF-8. Namely the input encodings in
1124     &#x2018;<tt>utf8.def</tt>&#x2019; from package
1125     <xref refid='inputenc'>inputenc</xref> and
1126     &#x2018;<tt>utf8x.def</tt>&#x2019; from package
1127     <xref refid='unicode'>ucs</xref> are supported.
1128     <p/>
1129     The package is part of the <xref refid='oberdiek'>oberdiek</xref>
1130     bundle.
1131   </description>
1132   <documentation details='Package documentation'
1133     href='ctan:/macros/latex/contrib/oberdiek/soulutf8.pdf' />
1134   <ctan file='true' path='/macros/latex/contrib/oberdiek/soulutf8.dtx' />
1135   <miktex location='oberdiek' />
1136   <texlive location='oberdiek' />
1137   <install path='/macros/latex/contrib/oberdiek/oberdiek.tds.zip' />
1138 </entry>
1139 </catalogue>
```

6 References

- [1] Melchior Franz: *The soul package*; 2003/11/17;
[CTAN:macros/latex/contrib/soul/](#).
- [2] Dominique P. G. Unruh: *ucs.sty – Unicode Support*; 2004/10/17;
[CTAN:macros/latex/contrib/unicode/](#).

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

7 History

[2007/09/09 v1.0]

- First version.

8 Index

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

Symbols		B	
\"	1074, 1078, 1098	\baselineskip	973
\#	656	\begin	897, 1068
\%	732	\bfseries	989
\@	657, 730	\body	674, 678
\@PackageError	148	\boxA	888, 902, 942, 943, 944, 945, 947, 950, 951, 952, 1003, 1004
\@PackageInfo	893, 898, 904, 911	\boxB	889, 909, 942, 944, 945, 948, 951, 1004, 1014
\@PackageWarningNoLine	185	\boxC	890, 923, 943, 950, 952
\@ehc	771	\break	245, 265, 267, 330, 350, 352
\@empty	899, 905, 912	C	
\@firstofone	665, 668	\c	1040, 1098
\@gobble	504, 662, 670	\catcode	2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 33, 34, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 69, 70, 72, 73, 74, 78, 79, 80, 81, 82, 83, 84, 87, 88, 90, 91, 92, 93, 97, 99, 169, 170, 172, 173, 654, 655, 656, 657, 692, 701, 709, 713, 730, 731, 732
\@ifDefined	767, 776, 783, 784	\cdot	838
\@ifdefinable	775	\char	816
\@ifundefined	768	\chardef	411, 438, 442, 446, 448
\@latex@error	769	\clearpage	1026
\@ne	215, 217, 219, 221, 300, 302, 304, 306, 931, 1051	\copy	965, 1003, 1014
\@nil	404, 407, 429, 433	\count@	659, 688, 692, 694, 695, 699, 701, 702, 703, 707, 709, 712, 713, 717, 718, 1044, 1046, 1047, 1051
\@onelevel@sanitiz	153, 381, 455, 471	\countdef	659
\@undefined	58, 129	\csname	14, 21, 50, 66, 76, 130, 153, 380, 381, 658, 661, 664, 667, 722, 749, 761, 770, 777, 778, 785, 786, 894, 1054
\\	242, 327, 731, 1029, 1090	\CurrentOption	162
\{	654, 975	D	
\}	655, 974	\DeclareOption	133, 138, 162
\~	1047		
_	816		
A			
\abc	1036, 1101, 1107		
\abx	1100, 1103, 1104		
\active	121		
\advance	215, 217, 219, 221, 300, 302, 304, 306, 695, 703, 718, 1051		
\afterassignment	178		
\aftergroup	29		
\an	814		
\AtBeginDocument	809		

<code>\DeclareRobustCommand</code>	131, 136, 814, 825, 850	<code>\IncludeTests</code>	886
<code>\DeclareUnicodeCharacter</code>	804	<code>\input</code>	141, 144, 152, 723
<code>\discretionary</code>	834, 844	<code>\iterate</code>	675, 677, 679
<code>\documentclass</code>	129, 756		
<code>\dp</code>	941, 951, 952		
		K	
		<code>\kern</code>	838
		L	
<code>\empty</code>	17, 18	<code>\lccode</code>	1047
<code>\end</code>	750, 953, 1109	<code>\leavevmode</code>	233, 318
<code>\endcsname</code>	14, 21, 50, 66, 76, 130, 153, 380, 381, 658, 661, 664, 667, 722, 749, 761, 770, 777, 778, 785, 786, 894, 1054	<code>\lefthyphenmin</code>	828
<code>\endinput</code>	29, 128	<code>\linebreak</code>	268, 270, 353, 355, 816, 818, 1028, 1091
<code>\endlinechar</code>	4, 35, 71, 77, 89	<code>\llap</code>	974
<code>\errmessage</code>	711	<code>\LoadCommand</code>	723, 733
<code>\etex@unexpanded</code>	464, 486, 494, 576, 601, 626, 645	<code>\LogTests</code>	887
<code>\Expect</code>	932, 933, 934, 940, 941, 942, 943, 944, 950, 951, 952	<code>\loop</code>	673, 689, 700, 708, 1045
		<code>\lowercase</code>	1048
		M	
<code>\gdef</code>	852, 880	<code>\makeatletter</code>	766, 813, 1024
		<code>\makeatother</code>	790, 1067
		<code>\maxdimen</code>	929, 930
		<code>\mbox</code>	255, 340, 1003, 1014, 1033, 1086, 1097
		<code>\meaning</code>	156, 402, 426
		<code>\MessageBreak</code>	186, 187, 188
		N	
<code>\hbox</code> ..	257, 342, 588, 600, 601, 613, 625, 626, 838, 843, 900, 907, 921	<code>\NeedsTeXFormat</code>	755
<code>\hfil</code>	245, 330	<code>\newcommand</code>	132, 137, 767, 774, 782, 878, 891, 955, 962, 1000, 1025
<code>\hfill</code>	966, 1006, 1015	<code>\newsavebox</code>	888, 889, 890
<code>\hfuzz</code>	958	<code>\next</code>	679, 681, 683
<code>\hsize</code>	957	<code>\nobreak</code>	245, 250, 330, 335, 459
<code>\HstringA</code>	901, 932, 933	<code>\nobreakspace</code>	804
<code>\HstringB</code>	908, 932	<code>\noindent</code>	964, 1002, 1013
<code>\HstringC</code>	922, 933	<code>\null</code>	230, 236, 244, 249, 315, 321, 329, 334, 458
<code>\ht</code>	940, 944, 945, 950, 1004	<code>\number</code>	714
		O	
		<code>\origshow</code>	807, 810
		P	
<code>\if</code>	197, 234, 282, 319	<code>\PackageError</code>	134, 139, 148
<code>\ifcase</code>	362, 365, 507, 515, 524, 532, 541, 549	<code>\PackageInfo</code>	26
<code>\ifcat</code>	271, 273, 356, 358	<code>\par</code>	232, 233, 317, 318, 963, 998, 1001, 1012, 1022
<code>\ifdim</code>	935, 945, 969, 970, 1004	<code>\PassOptionsToPackage</code>	162
<code>\ifetex@unexpanded</code>	168	<code>\phantom</code> ..	976, 982, 991, 996, 1010, 1019
<code>\ifnum</code>	201, 203, 205, 207, 286, 288, 290, 292, 694, 702, 709, 717, 972, 980, 1046	<code>\ProcessOptions</code>	163
<code>\ifSOUL@ignorespaces</code>	237, 322	<code>\protect</code> ..	559, 571, 584, 596, 609, 621
<code>\ifx</code>	15, 18, 21, 50, 58, 61, 129, 130, 153, 182, 210, 211, 212, 213, 214, 216, 218, 220, 227, 232, 242, 247, 251, 255, 257, 259, 265, 268, 295, 296, 297, 298, 299, 301, 303, 305, 312, 317, 327, 332, 336, 340, 342, 344, 350, 353, 412, 414, 416, 437, 441, 445, 456, 472, 478, 658, 661, 664, 667, 722	<code>\ProvidesPackage</code>	19, 67
<code>\immediate</code>	23, 52		
		Q	
		<code>\qqquad</code>	977, 983, 993
		R	
		<code>\raisebox</code>	973

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

S
`\SavedUnexpanded` 753, 801
`\SaveOrg` 774, 791, 792,
793, 794, 795, 796, 797, 798, 799
`\section` 1055
`\setbox` 900, 907, 921, 956
`\SetTestVBox` 902, 909, 923, 955
`\show` 807, 810
`\showbox` 937, 938, 947, 948
`\showboxbreadth` 930
`\showboxdepth` 929
`\slash` 251, 336, 1089
`\small` 968, 1008, 1017
`\SOUL@` 823, 848, 876
`\SOUL@@` 197, 210, 211,
212, 213, 214, 216, 218, 220,
227, 232, 234, 242, 247, 251,
255, 257, 259, 265, 268, 271,
273, 282, 295, 296, 297, 298,
299, 301, 303, 305, 312, 317,
319, 327, 332, 336, 340, 342,
344, 350, 353, 356, 358, 402, 426
`\SOUL@addfouroctets` 373, 492
`\SOUL@addgroup` 272, 357, 557
`\SOUL@addmath` 274, 359, 582
`\SOUL@addprotect` 256, 258, 341, 343, 607
`\SOUL@addthreeoctets` 371, 469
`\SOUL@addtoken`
. 463, 464, 485, 486, 493, 494, 632
`\SOUL@addtwooctets` 369, 453
`\SOUL@analyzeutfviii` 361, 398
`\SOUL@analyzeutfviiix` 363, 422
`\SOUL@apo` 205,
219, 290, 304, 524, 530, 532, 538
`\SOUL@AtEnd` 95, 96, 128, 651
`\SOUL@charhash` 394, 427, 432
`\SOUL@charkern` 845
`\SOUL@charthree` 396, 441
`\SOUL@chartwo` 395, 445
`\SOUL@checkutfviii` 401, 406
`\SOUL@checkutfviiix` 425, 431
`\SOUL@cmd` 177, 182, 184, 193
`\SOUL@cmdcheck` 178, 181
`\SOUL@comma` 203,
217, 288, 302, 507, 513, 515, 521
`\SOUL@defsanitizedstring` 379, 383,
384, 385, 386, 387, 388, 389, 390
`\SOUL@dotoken` 276, 366
`\SOUL@doword`
. 229, 235, 243, 248, 252, 261,
266, 269, 314, 320, 328, 333,
337, 346, 351, 354, 457, 473, 479
`\SOUL@empty` 397, 437
`\SOUL@error` 144
`\SOUL@eval` 195
`\SOUL@eventuallyexhyphen`
. 230, 236, 244, 249, 253, 315,
321, 329, 334, 338, 458, 474, 480
`\SOUL@everyexhyphen` ... 819, 841, 864
`\SOUL@everyhyphen` ... 818, 833, 861
`\SOUL@everylowerthan` 822, 873
`\SOUL@everyspace`
..... 239, 245, 250, 270, 324,
330, 335, 355, 459, 816, 831, 855
`\SOUL@everysyllable` ... 820, 832, 867
`\SOUL@everytoken` 821, 870
`\SOUL@exhyphen` ... 254, 339, 475, 481
`\SOUL@flushapo`
. 206, 212, 225, 291, 297, 310, 523
`\SOUL@flushcomma`
. 204, 211, 224, 289, 296, 309, 506
`\SOUL@flushgrave`
. 208, 213, 226, 293, 298, 311, 540
`\SOUL@flushminus`
..... 202, 210, 223, 287, 295, 308
`\SOUL@grave` 207,
221, 292, 306, 541, 547, 549, 555
`\SOUL@hyphkern` 835, 842
`\SOUL@ignorespacesfalse` ... 199, 284
`\SOUL@ignorespacestrue` ... 246, 331
`\SOUL@minus` 201, 215, 286, 300
`\SOUL@n` 196, 228, 233, 241, 256, 258,
260, 272, 274, 276, 279, 281,
313, 318, 326, 341, 343, 345,
357, 359, 366, 369, 371, 373, 377
`\SOUL@next`
. 460, 462, 467, 476, 482, 484, 490
`\SOUL@octets`
. 362, 365, 411, 438, 442, 446, 448
`\SOUL@orgDeclareOption` ... 133, 138
`\SOUL@orgDeclareRobustCommand` ..
..... 131, 136
`\SOUL@orgnewcommand` 132, 137
`\SOUL@orgPackageError` 134, 139
`\SOUL@orgSOUL@error` 145, 149
`\SOUL@postamble` 817, 858
`\SOUL@preamble` 815, 827, 851
`\SOUL@redefine` ... 175, 195, 496,
506, 523, 540, 557, 582, 607, 632
`\SOUL@remainingtoken` 501, 503
`\SOUL@restorelatexcmds` ... 135, 142
`\SOUL@scan`
. 196, 233, 241, 263, 281, 318,

[illegible]