

The soulutf8 package

Heiko Oberdiek
<oberdiek@uni-freiburg.de>

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	3
2.3	Loading packages	4
2.3.1	plain-TeX	4
2.3.2	L ^A T _E 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
3	Test	14
3.1	Catcode checks for loading	14
3.2	L ^A T _E Xtests	16
3.2.1	Driver definitions	17
3.2.2	Test macros	18
3.2.3	Testing	21
4	Installation	22
4.1	Download	22
4.2	Bundle installation	22
4.3	Package installation	23
4.4	Refresh file name databases	23
4.5	Some details for the interested	23
5	References	24

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

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
3 \catcode44 12 % ,
4 \catcode45 12 % -
5 \catcode46 12 % .
6 \catcode58 12 % :
7 \catcode64 11 % @
8 \expandafter\let\expandafter\x\csname ver@soulutf8.sty\endcsname
9 \ifcase 0%
10 \ifx\x\relax % plain
11 \else
```

```

12     \ifx\x\empty % LaTeX
13     \else
14         1%
15     \fi
16 \fi
17 \else
18     \catcode35 6 % #
19     \catcode123 1 % {
20     \catcode125 2 % }
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     \endgroup
30     \expandafter\endinput
31 \fi
32 \endgroup

```

Package identification:

```

33 \begingroup
34 \catcode35 6 % #
35 \catcode40 12 % (
36 \catcode41 12 % )
37 \catcode44 12 % ,
38 \catcode45 12 % -
39 \catcode46 12 % .
40 \catcode47 12 % /
41 \catcode58 12 % :
42 \catcode64 11 % @
43 \catcode123 1 % {
44 \catcode125 2 % }
45 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
46     \def\x#1#2#3[#4]{\endgroup
47         \immediate\write-1{Package: #3 #4}%
48         \xdef#1{#4}%
49     }%
50 \else
51     \def\x#1#2[#3]{\endgroup
52         #2[#{#3}]%
53         \ifx#1\@undefined
54             \xdef#1{#3}%
55         \fi
56         \ifx#1\relax
57             \xdef#1{#3}%
58         \fi
59     }%
60 \fi
61 \expandafter\x\csname ver@soulutf8.sty\endcsname
62 \ProvidesPackage{soulutf8}%
63 [2007/09/09 v1.0 Adding support for UTF-8 to soul (HO)]

```

2.2 Catcodes

```

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

```

```

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

```

2.3 Loading packages

Package soul uses \documentclass to detect L^AT_EX.

```

113 \ifx\documentclass\@undefined

```

2.3.1 plain-T_EX

First we check, whether package soul is already loaded.

```

114 \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 miniltx.tex, for example. Therefore these macros are first saved and restored afterwards.

```

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

```

```

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

```

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

```

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

136  \input etexcmds.sty\relax

```

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

```

137  \expandafter\ifx\csname @onelevel@sanitize\endcsname\relax
138    \def\@onelevel@sanitize#1{%
139      \edef#1{%
140        \expandafter\strip@prefix\meaning#1%
141      }%
142    }%

```

\strip@prefix

```

143    \def\strip@prefix#1>{}%

144  \fi
145  \else

```

2.3.2 L^AT_EX

```

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

```

2.3.3 ε-T_EX

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

```

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

```

2.4 Macro for redefinitions

\SOuL@redefine

```

159  \def\SOuL@redefine#1{%
160    \begingroup
161    \def\SOuL@cmd{#1}%

```

```

162     \afterassignment\SOuL@cmdcheck
163     \def\SOuL@temp
164 }

\SOuL@cmdcheck

165 \def\SOuL@cmdcheck{%
166     \expandafter\ifx\SOuL@cmd\SOuL@temp
167     \else
168         \edef\SOuL@temp*{\expandafter\string\SOuL@cmd}%
169         \@PackageWarningNoLine{soulutf8}{%
170             Command \SOuL@temp* has changed.\MessageBreak
171             Supported versions of package 'soul': 2003/11/17.\MessageBreak
172             Depending on the unknown changes the redefinition\MessageBreak
173             of \SOuL@temp* may not behave correctly%
174         }%
175     \fi
176 \expandafter\endgroup
177 \expandafter\def\SOuL@cmd
178 }

```

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.

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

First the expected definition.

```

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

```

```

214     \SOUL@eventuallyexhyphen\null
215 }%
216 \else\ifx\SOUL@@\par
217     \def\SOUL@n*\par{\par\leavevmode\SOUL@scan}%
218 \else\if\noexpand\SOUL@@\SOUL@spc
219     \SOUL@doword
220     \SOUL@eventuallyexhyphen\null
221     \ifSOUL@ignorespaces
222     \else
223         \SOUL@everySPACE{}%
224     \fi
225     \def\SOUL@n* {\SOUL@scan}%
226 \else\ifx\SOUL@@\%
227     \SOUL@doword
228     \SOUL@eventuallyexhyphen\null
229     \SOUL@everySPACE{\unskip\nobreak\hfil\break}%
230     \SOUL@ignorespacestrue
231 \else\ifx\SOUL@@~%
232     \SOUL@doword
233     \SOUL@eventuallyexhyphen\null
234     \SOUL@everySPACE{\nobreak}%
235 \else\ifx\SOUL@@\slash
236     \SOUL@doword
237     \SOUL@eventuallyexhyphen{/}%
238     \SOUL@exhyphen{/}%
239 \else\ifx\SOUL@@\mbox
240     \def\SOUL@n*{\SOUL@addprotect}%
241 \else\ifx\SOUL@@\hbox
242     \def\SOUL@n*{\SOUL@addprotect}%
243 \else\ifx\SOUL@@\soulomit
244     \def\SOUL@n*\soulomit##1{%
245         \SOUL@doword
246         {\spaceskip\SOUL@spaceskip##1}%
247         \SOUL@scan
248     }%
249 \else\ifx\SOUL@@\break
250     \SOUL@doword
251     \break
252 \else\ifx\SOUL@@\linebreak
253     \SOUL@doword
254     \SOUL@everySPACE{\linebreak}%
255 \else\ifcat\bgroup\noexpand\SOUL@@
256     \def\SOUL@n*{\SOUL@addgroup{}}%
257 \else\ifcat$\noexpand\SOUL@@
258     \def\SOUL@n*{\SOUL@addmath}%
259 \else
260     \def\SOUL@n*{\SOUL@dotoken}%
261     \fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi
262 \fi\fi\fi\fi
263 \SOUL@n*%
264 }{%

```

Now the redefined version follows.

```

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

```

```

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

```



```

390 \def\noexpand\SouL@checkutfviii
391   ##1\SouL@stringUTFviii##2\SouL@stringoctets##3\noexpand\@nil
392 }%
393 \x{%
394   \def\SouL@temp{#2}%
395   \chardef\SouL@octets=%
396     \ifx\SouL@temp\SouL@stringtwo
397       \tw@
398     \else\ifx\SouL@temp\SouL@stringthree
399       \thr@@
400     \else\ifx\SouL@temp\SouL@stringfour
401       4 %
402     \else
403       \z@
404     \fi\fi\fi
405 }

```

2.6.3 Support for utf8x.def

\SouL@analyzeutfviii

```

406 \begingroup
407 \edef\x{\endgroup
408   \def\noexpand\SouL@analyzeutfviii{%
409     \noexpand\expandafter\noexpand\SouL@checkutfviii
410     \noexpand\meaning\noexpand\SouL@@
411     \SouL@stringmacrocolon\SouL@charhash1{}{}{}{}%
412     \SouL@stringcsnameu\SouL@stringundeferr
413     \noexpand\@nil
414   }%

```

\SouL@checkutfviii

```

415 \def\noexpand\SouL@checkutfviii
416   ##1\SouL@stringmacrocolon\SouL@charhash1##2##3##4##5##6%
417   \SouL@stringcsnameu##7\SouL@stringundeferr##8\noexpand\@nil
418 }%
419 \x{%
420   \def\SouL@temp{#7}%
421   \ifx\SouL@temp\SouL@empty
422     \chardef\SouL@octets=\z@
423   \else
424     \def\SouL@temp{#5}%
425     \ifx\SouL@temp\SouL@charthree
426       \chardef\SouL@octets=4 %
427     \else
428       \def\SouL@temp{#3}%
429       \ifx\SouL@temp\SouL@chartwo
430         \chardef\SouL@octets=\thr@@
431       \else
432         \chardef\SouL@octets=\tw@
433       \fi
434     \fi
435   \fi
436 }

```

2.7 Actions for UTF-8 sequences

\SouL@addtwooctets

```

437 \def\SouL@addtwooctets#1#2{%
438   \def\SouL@temp{#1#2}%
439   \@onelevel@sanitize\SouL@temp
440   \ifx\SouL@temp\SouL@stringnobreakspace

```

```

441 \SOUL@doword
442 \SOUL@eventuallyexhyphen\null
443 \SOUL@everySPACE{\nobreak}%
444 \let\SOuL@next\SOUL@scan
445 \else
446 \def\SOuL@next{%
447 ! \SOUL@addtoken{\noexpand#1\noexpand#2}}%
448 + \SOUL@addtoken{\etex@unexpanded{#1#2}}}%
449 }%
450 \fi
451 \SOuL@next
452 }

```

\SOuL@addthreeoctets

```

453 \def\SOuL@addthreeoctets#1#2#3{%
454 \def\SOuL@temp{#1#2#3}%
455 \@onelevel@sanitize\SOuL@temp
456 \ifx\SOuL@temp\SOuL@stringendash
457 \SOUL@doword
458 \SOUL@eventuallyexhyphen{-}%
459 \SOUL@exhyphen{--}%
460 \let\SOuL@next\SOUL@scan
461 \else
462 \ifx\SOuL@temp\SOuL@stringemdash
463 \SOUL@doword
464 \SOUL@eventuallyexhyphen{-}%
465 \SOUL@exhyphen{---}%
466 \let\SOuL@next\SOUL@scan
467 \else
468 \def\SOuL@next{%
469 ! \SOUL@addtoken{\noexpand#1\noexpand#2\noexpand#3}}%
470 + \SOUL@addtoken{\etex@unexpanded{#1#2#3}}}%
471 }%
472 \fi
473 \fi
474 \SOuL@next
475 }

```

\SOuL@addfouroctets

```

476 \def\SOuL@addfouroctets#1#2#3#4{%
477 ! \SOUL@addtoken{\noexpand#1\noexpand#2\noexpand#3\noexpand#4}}%
478 + \SOUL@addtoken{\etex@unexpanded{#1#2#3#4}}}%
479 }

```

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.

```

480 \SOUL@redefine\SOUL@splittoken#1#2\SOUL@stop{%
481 \global\SOUL@token={#1}%
482 \global\SOUL@word={#2}%
483 }#1{%
484 \global\SOUL@token={#1}%
485 \SOuL@remainingtoken\relax
486 }

```

\SOuL@remainingtoken

```

487 \def\SOuL@remainingtoken#1\SOUL@stop{%

```

```

488 \global\SOUL@word=\expandafter{\@gobble#1}%
489 }

```

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`

```

490 \SOUL@redefine\SOUL@flushcomma{%
491 \ifcase\SOUL@comma
492 \or
493 \edef\x{\SOUL@word={\the\SOUL@word,}}\x
494 \or
495 \edef\x{\SOUL@word={\the\SOUL@word{, ,}}}\x
496 \fi
497 \SOUL@comma\z@
498 }{%
499 \ifcase\SOUL@comma
500 \or
501 \edef\x{\SOUL@word={\the\SOUL@word,}}\x
502 \or
503 \edef\x{\SOUL@word={\the\SOUL@word{, ,}}}\x
504 \fi
505 \SOUL@comma\z@
506 }

```

`\SOUL@flushapo`

```

507 \SOUL@redefine\SOUL@flushapo{%
508 \ifcase\SOUL@apo
509 \or
510 \edef\x{\SOUL@word={\the\SOUL@word'}}\x
511 \or
512 \edef\x{\SOUL@word={\the\SOUL@word{' '}}}\x
513 \fi
514 \SOUL@apo\z@
515 }{%
516 \ifcase\SOUL@apo
517 \or
518 \edef\x{\SOUL@word={\the\SOUL@word'}}\x
519 \or
520 \edef\x{\SOUL@word={\the\SOUL@word{' '}}}\x
521 \fi
522 \SOUL@apo\z@
523 }

```

`\SOUL@flushgrave`

```

524 \SOUL@redefine\SOUL@flushgrave{%
525 \ifcase\SOUL@grave
526 \or
527 \edef\x{\SOUL@word={\the\SOUL@word'}}\x
528 \or
529 \edef\x{\SOUL@word={\the\SOUL@word{' '}}}\x
530 \fi
531 \SOUL@grave\z@
532 }{%
533 \ifcase\SOUL@grave
534 \or
535 \edef\x{\SOUL@word={\the\SOUL@word'}}\x
536 \or

```

```

537 \edef\x{\SOUL@word={\the\SOUL@word{'}}}\x
538 \fi
539 \SOUL@grave\z@
540 }

```

\SOUL@addgroup

```

541 \SOuL@redefine\SOUL@addgroup#1#2{%
542 {%
543 \let\protect\noexpand
544 \edef\x{%
545 \global\SOUL@word={%
546 \the\SOUL@word
547 {\noexpand#1#2}}%
548 }%
549 }%
550 \x
551 }%
552 \SOUL@scan
553 }#1#2{%
554 \begingroup
555 \let\protect\noexpand
556 \edef\x{\endgroup
557 \SOUL@word={%
558 \the\SOUL@word
559 ! {\noexpand#1#2}}%
560 + {\etex@unexpanded{#1#2}}}%
561 }%
562 }%
563 \x
564 \SOUL@scan
565 }

```

\SOUL@addmath

```

566 \SOuL@redefine\SOUL@addmath$#1${%
567 {%
568 \let\protect\noexpand
569 \edef\x{%
570 \global\SOUL@word={%
571 \the\SOUL@word
572 {\hbox{ $#1$ }}%
573 }%
574 }%
575 \x
576 }%
577 \SOUL@scan
578 }$#1${%
579 \begingroup
580 \let\protect\noexpand
581 \edef\x{\endgroup
582 \SOUL@word={%
583 \the\SOUL@word
584 ! {\hbox{ $#1$ }}%
585 + {\etex@unexpanded{\hbox{ $#1$ }}}%
586 }%
587 }%
588 \x
589 \SOUL@scan
590 }

```

\SOUL@addprotect

```

591 \SOuL@redefine\SOUL@addprotect#1#2{%
592 {%

```

```

593 \let\protect\noexpand
594 \edef\x{%
595   \global\SOUL@word={%
596     \the\SOUL@word
597     {\hbox{#2}}}%
598   }%
599 }%
600 \x
601 }%
602 \SOUL@scan
603 }#1#2{%
604   \begingroup
605     \let\protect\noexpand
606     \edef\x{\endgroup
607       \SOUL@word={%
608         \the\SOUL@word
609         {\hbox{#2}}}%
610       {\etex@unexpanded{\hbox{#2}}}%
611     }%
612   }%
613   \x
614   \SOUL@scan
615 }

```

\SOUL@addtoken

```

616 + \SOUL@redefine\SOUL@addtoken#1{%
617 +   \edef\x{%
618 +     \SOUL@word={%
619 +       \the\SOUL@word
620 +       \noexpand#1%
621 +     }%
622 +   }%
623 +   \x
624 +   \SOUL@scan
625 + }#1{%
626 +   \edef\x{%
627 +     \SOUL@word={%
628 +       \the\SOUL@word
629 +       \etex@unexpanded{#1}%
630 +     }%
631 +   }%
632 +   \x
633 +   \SOUL@scan
634 + }%

635 \SOUL@AtEnd
636 </package>

```

3 Test

3.1 Catcode checks for loading

```

637 <*test1>

638 \catcode'\{=1 %
639 \catcode'\}=2 %
640 \catcode'\#=6 %
641 \catcode'\@=11 %
642 \expandafter\ifx\csname count@\endcsname\relax
643 \countdef\count@=255 %
644 \fi
645 \expandafter\ifx\csname @gobble\endcsname\relax

```

```

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

```

```

708 \LoadCommand
709 \RestoreCatcodes
710 }
711 \Test
712 \csname @@end\endcsname
713 \end
714 </test1>

```

3.2 L^AT_EX tests

```

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

```

Input encoding utf8.

```

765 <*utf8>

```



```

766 \usepackage[utf8]{inputenc}
767 \DeclareUnicodeCharacter{00A0}{\nobreakspace}
768 \end{utf8}

```

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

```

769 \begin{utf8x}
770 \let\origshow\show
771 \usepackage[utf8x]{inputenc}
772 \AtBeginDocument{%
773   \let\show\origshow
774 }
775 \end{utf8x}

```

3.2.1 Driver definitions

```

776 \makeatletter

```

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

```

777 \DeclareRobustCommand*\an{%
778   \def\SOUL@preamble{$\sim P}$}%
779   \def\SOUL@everyspace##1{##1\texttt{\char'\ }\\linebreak[2]}%
780   \def\SOUL@postamble{$\sim E}$}%
781   \def\SOUL@everyhyphen{$\sim$}\\linebreak[1]}%
782   \def\SOUL@everyexhyphen##1{##1$\sim$}%
783   \def\SOUL@everysyllable{$\sim S}$}%
784   \def\SOUL@everytoken{\the\SOUL@token$\sim T}$}%
785   \def\SOUL@everylowerthan{$\sim L}$}%
786   \SOUL@
787 }

```

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

```

788 \DeclareRobustCommand*\sy{%
789   \SOUL@setup
790   \def\SOUL@preamble{%
791     \lefthyphenmin=\z@
792     \righthyphenmin=\z@
793   }%
794   \def\SOUL@everyspace##1{##1\space}%
795   \def\SOUL@everysyllable{\the\SOUL@syllable}%
796   \def\SOUL@everyhyphen{%
797     \discretionary{%
798       \SOUL@setkern\SOUL@hyphkern
799       \SOUL@sethyphenchar
800     }{}{%
801       \hbox{\kern1pt$\cdot$}%
802     }%
803   }%
804   \def\SOUL@everyexhyphen##1{%
805     \SOUL@setkern\SOUL@hyphkern
806     \hbox{##1}%
807     \discretionary{}{}{%
808       \SOUL@setkern\SOUL@charkern
809     }%
810   }%
811   \SOUL@
812 }

```

`\soulH`

```

813 \DeclareRobustCommand*\soulH{%
814   \def\SOUL@preamble{%
815     \gdef\SOULH{}%
816     \soulHadd{}%
817   }%
818   \def\SOUL@everyspace##1{%

```

```

819 \soulHadd{Sp}%
820 }%
821 \def\Soul@postamble{%
822 \soulHadd{}}%
823 }%
824 \def\Soul@everyhyphen{%
825 \soulHadd{Hy}%
826 }%
827 \def\Soul@everyexhyphen##1{%
828 \soulHadd{Exhy}%
829 }%
830 \def\Soul@everysyllable{%
831 \soulHadd{Sy}%
832 }%
833 \def\Soul@everytoken{%
834 \soulHadd{T}%
835 }%
836 \def\Soul@everylowerthan{%
837 \soulHadd{<}%
838 }%
839 \Soul@
840 }
841 \newcommand*{\soulHadd}[1]{%
842 #1%
843 \expandafter\gdef\expandafter\soulHstring\expandafter{%
844 \soulHstring
845 #1%
846 }%
847 }

```

3.2.2 Test macros

```

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

851 \newsavebox\boxA
852 \newsavebox\boxB
853 \newsavebox\boxC

\TestDriver

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

```

```

876 \RestoreOrg{splittoken}%
877 \RestoreOrg{flushcomma}%
878 \RestoreOrg{flushapo}%
879 \RestoreOrg{grave}%
880 \RestoreOrg{addgroup}%
881 \RestoreOrg{addmath}%
882 \RestoreOrg{addprotect}%
883 \RestoreOrg{addtoken}%
884 \setbox4=\hbox{\expandafter\testdriver\expandafter{\teststringB}}%
885 \let\HstringC\soulHstring
886 \SetTestVBox{\boxC}{\teststringB}%
887 %
888 \TestInfo{0}{soulutf8}%
889 \TestInfo{2}{soul + patch}%
890 \TestInfo{4}{original soul}%
891 \TestInfoVBox
892 \showboxdepth=\maxdimen
893 \showboxbreadth=\maxdimen
894 \tracingonline=@ne
895 \Expect*{\HstringA}*{\HstringB}%
896 \Expect*{\HstringA}*{\HstringC}%
897 \Expect*{\the\wd0}*{\the\wd2}%
898 \ifdim\wd0=\wd2 %
899 \else
900 \showbox0 %
901 \showbox2 %
902 \fi
903 \Expect*{\the\ht0}*{\the\ht2}%
904 \Expect*{\the\dp0}*{\the\dp2}%
905 \Expect*{\the\wd\boxB}*{\the\wd\boxA}%
906 \Expect*{\the\wd\boxC}*{\the\wd\boxA}%
907 \Expect*{\the\ht\boxB}*{\the\ht\boxA}%
908 \ifdim\ht\boxB=\ht\boxA
909 \else
910 \showbox\boxA
911 \showbox\boxB
912 \fi
913 \Expect*{\the\ht\boxC}*{\the\ht\boxA}%
914 \Expect*{\the\dp\boxB}*{\the\dp\boxA}%
915 \Expect*{\the\dp\boxC}*{\the\dp\boxA}%
916 \end{qstest}%
917 }

```

\SetTestVBox

```

918 \newcommand*\SetTestVBox}[2]{%
919 \setbox#1=\vbox{%
920 \hsize=0pt\relax
921 \hfuzz=100mm\relax
922 \expandafter\testdriver\expandafter{#2}%
923 }%
924 }

```

\TestInfo

```

925 \newcommand*\TestInfo}[2]{%
926 \par
927 \noindent
928 \copy#1\relax
929 \hfill
930 \begin{group}
931 \small
932 \ifdim\wd0=\wd2\relax
933 \ifdim\wd0=\wd4\relax

```

```

934     \else
935     \ifnum#1=2 %
936     \raisebox{.5\baselineskip}[0pt][0pt]{%
937     \llap{$\}$ }%
938     \rlap{width = \the\wd#1\relax\space$\{ $\}%
939     \phantom{width = 999.99999pt}}%
940     \qqquad
941     }%
942     \else
943     \ifnum#1=4 %
944     \rlap{width = \the\wd#1}%
945     \phantom{width = 999.99999pt}}%
946     \qqquad
947     \fi
948     \fi
949     \fi
950     \else
951     \begingroup
952     \bfseries
953     \rlap{width = \the\wd#1}%
954     \phantom{width = 999.99999pt}}%
955     \endgroup
956     \qqquad
957     \fi
958     \rlap{(#2)}%
959     \phantom{(original soul)}}%
960 \endgroup
961 \par
962 }

```

\TestInfoVBox

```

963 \newcommand*{\TestInfoVBox}{%
964 \par
965 \noindent
966 \mbox{\vrule\relax\space\copy\boxA}%
967 \ifdim\ht\boxA=\ht\boxB
968 \else
969 \hfill
970 \begingroup
971 \small
972 \rlap{(soulutf8)}%
973 \phantom{(soulutf8)}}%
974 \endgroup
975 \par
976 \noindent
977 \mbox{\vrule\relax\space\copy\boxB}%
978 \hfill
979 \begingroup
980 \small
981 \rlap{(soul)}%
982 \phantom{(soulutf8)}}%
983 \endgroup
984 \fi
985 \par
986 }

```

\Test

```

987 \makeatletter
988 \newcommand*{\Test}[3]{%
989 \clearpage
990 \begingroup
991 \def\linebreak{\textbackslash linebreak }%

```

```

992 \def\{\textbackslash\textbackslash}%
993 \def\soulomit##1{%
994 \textbackslash soulomit\textbraceleft##1\textbraceright
995 }%
996 \def\mbox##1{%
997 \textbackslash mbox\textbraceleft##1\textbraceright
998 }%
999 \def\abc{\textbackslash abc }%
1000 \def\texttt##1{%
1001 \textbackslash texttt\textbraceleft##1\textbraceright
1002 }%
1003 \def\c##1{%
1004 \textbackslash c\textbraceleft##1\textbraceright
1005 }%
1006 <*utf8x>
1007 \count@=128 %
1008 \loop
1009 \ifnum\count@<256 %
1010 \lccode'\~=\count@
1011 \lowercase{%
1012 \def~{\noexpand~}%
1013 }%
1014 \advance\count@\@ne
1015 \repeat
1016 </utf8x>
1017 \csname protected@edef\endcsname\x{\endgroup
1018 \noexpand\section{String #1: #2}%
1019 }%
1020 \x
1021 \TestDriver{so}{#1}{#2}{#3}%
1022 \TestDriver{caps}{#1}{#2}{#3}%
1023 \TestDriver{ul}{#1}{#2}{#3}%
1024 \TestDriver{st}{#1}{#2}{#3}%
1025 \TestDriver{hl}{#1}{#2}{#3}%
1026 \TestDriver{an}{#1}{#2}{#3}%
1027 \TestDriver{sy}{#1}{#2}{#3}%
1028 \TestDriver{soulH}{#1}{#2}{#3}%
1029 }
1030 \makeatother

```

3.2.3 Testing

```

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

```

```

1052 \Test Q{re-sent\slash x}{re-sent\slash x}%
1053 \Test R{new\\line}{new\\line}%
1054 \Test S{foo\linebreak bar}{foo\linebreak bar}%
1055 \Test T{Virchow{sche}}{Virchow{sche}}%
1056 \Test U{A\soulomit{foo}B}{A\soulomit{foo}B}%
1057 \Test V{A\texttt{xyz}B}{A\texttt{xyz}B}%
1058 \Test W{ffi}{ffi}%
1059 \Test X{f{}f{}i}{f{}f{}i}%
1060 \Test Y{f\mbox{}f\mbox{}i}{f\mbox{}f\mbox{}i}%
1061 \Test Z{\c{"a}}{\c{"a}}%
1062 <*test2 | test3>
1063 \let\abcx\UNDEFINED
1064 \def\abc{%
1065   \begingroup
1066     \def\abcx{x}%
1067     \abcx
1068   \endgroup
1069 }%
1070 \Test{AA}{[{x\abc x}]}[{{xxx}}]%
1071 </test2 | test3>
1072 \end{document}
1073 </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 \TeX 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/
```

¹<http://ftp.ctan.org/tex-archive/>

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 `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	
\ "	1037, 1041, 1061
\ #	640, 702
\ %	705
\ @	641, 698
\@PackageError	132
\@PackageInfo	856, 861, 867, 874
\@PackageWarningNoLine	169
\@ehc	734
\@empty	862, 868, 875
\@firstofone	649, 652
\@gobble	488, 646, 654
\@ifDefined	730, 739, 746, 747
\@ifdefinable	738
\@ifundefined	731
\@latex@error	732
\@ne	199, 201, 203,
	205, 284, 286, 288, 290, 894, 1014
\@nil	388, 391, 413, 417
\@onelevel@sanitize	<u>137</u> , 365, 439, 455
\@undefined	53, 113
\[.....	703
\\	226, 311, 699, 992, 1053
\{	638, 700, 938
\}	639, 701, 937
\]	704
\~	1010
_	706, 779
A	
\abc	999, 1064, 1070
\abcx	1063, 1066, 1067
\active	106
\advance	199, 201, 203, 205,
	284, 286, 288, 290, 679, 687, 1014
B	
\afterassignment	162
\an	<u>777</u>
\AtBeginDocument	772
C	
\c	1003, 1061
\catcode	3, 4, 5, 6, 7, 18,
	19, 20, 34, 35, 36, 37, 38, 39, 40,
	41, 42, 43, 44, 65, 66, 69, 70, 71,
	72, 76, 77, 78, 79, 83, 85, 153,
	154, 156, 157, 638, 639, 640,
	641, 676, 685, 698, 699, 700,
	701, 702, 703, 704, 705, 706, 707
\cdot	801
\char	779
\chardef	395, 422, 426, 430, 432
\clearpage	989
\copy	928, 966, 977
\count@	643,
	672, 676, 678, 679, 683, 685,
	686, 687, 1007, 1009, 1010, 1014
\countdef	643
\csname	8, 21, 45,
	61, 68, 114, 137, 364, 365, 642,

645, 648, 651, 690, 712, 724, 733, 740, 741, 748, 749, 857, 1017	\input 125, 128, 136, 691
\CurrentOption 146	\iterate 659, 661, 663
D	K
\DeclareOption 117, 122, 146	\kern 801
\DeclareRobustCommand 115, 120, 777, 788, 813	L
\DeclareUnicodeCharacter 767	\lccode 1010
\discretionary 797, 807	\leavevmode 217, 302
\documentclass 113, 719	\lefthyphenmin 791
\dp 904, 914, 915	\linebreak 252, 254, 337, 339, 779, 781, 991, 1054
E	\llap 937
\empty 12	\LoadCommand 691, 708
\end 713, 916, 1072	\LogTests 850
\endcsname 8, 21, 45, 61, 68, 114, 137, 364, 365, 642, 645, 648, 651, 690, 712, 724, 733, 740, 741, 748, 749, 857, 1017	\loop 657, 673, 684, 1008
\endinginput 30	\lowercase 1011
\etex@unexpanded 448, 470, 478, 560, 585, 610, 629	M
\Expect 895, 896, 897, 903, 904, 905, 906, 907, 913, 914, 915	\makeatletter 729, 776, 987
G	\makeatother 753, 1030
\gdef 815, 843	\maxdimen 892, 893
H	\mbox 239, 324, 966, 977, 996, 1049, 1060
\hbox .. 241, 326, 572, 584, 585, 597, 609, 610, 801, 806, 863, 870, 884	\meaning 140, 386, 410
\hfil 229, 314	\MessageBreak 170, 171, 172
\hfill 929, 969, 978	N
\hfuzz 921	\NeedsTeXFormat 718
\hsize 920	\newcommand 116, 121, 730, 737, 745, 841, 854, 918, 925, 963, 988
\HstringA 864, 895, 896	\newsavebox 851, 852, 853
\HstringB 871, 895	\next 663, 665, 667
\HstringC 885, 896	\nobreak 229, 234, 314, 319, 443
\ht 903, 907, 908, 913, 967	\nobreakspace 767
I	\noindent 927, 965, 976
\if 181, 218, 266, 303	\null 214, 220, 228, 233, 299, 305, 313, 318, 442
\ifcase 9, 346, 349, 491, 499, 508, 516, 525, 533	O
\ifcat 255, 257, 340, 342	\origshow 770, 773
\ifdim 898, 908, 932, 933, 967	P
\ifetex@unexpanded 152	\PackageError 118, 123, 132
\ifnum . 185, 187, 189, 191, 270, 272, 274, 276, 678, 686, 935, 943, 1009	\PackageInfo 26
\ifSOUL@ignorespaces 221, 306	\par 216, 217, 301, 302, 926, 961, 964, 975, 985
\ifx 10, 12, 21, 45, 53, 56, 113, 114, 137, 166, 194, 195, 196, 197, 198, 200, 202, 204, 211, 216, 226, 231, 235, 239, 241, 243, 249, 252, 279, 280, 281, 282, 283, 285, 287, 289, 296, 301, 311, 316, 320, 324, 326, 328, 334, 337, 396, 398, 400, 421, 425, 429, 440, 456, 462, 642, 645, 648, 651, 690	\PassOptionsToPackage 146
\immediate 23, 47	\phantom .. 939, 945, 954, 959, 973, 982
\IncludeTests 849	\ProcessOptions 147
	\protect .. 543, 555, 568, 580, 593, 605
	\ProvidesPackage 62
	Q
	\qqquad 940, 946, 956
	R
	\raisebox 936
	\RangeCatcodeInvalid 682, 694, 695, 696, 697
	\repeat 657, 669, 680, 688, 1015
	\RequirePackage 148, 149, 150
	\RestoreCatcodes .. 671, 674, 675, 709
	\RestoreOrg 745, 869, 876, 877, 878, 879, 880, 881, 882, 883

\righthyphenmin	792	\SOUL@everylowerthan	785, 836
\rlap	938, 944, 953, 958, 972, 981	\SOUL@everyospace	223, 229, 234, 254, 308, 314, 319, 339, 443, 779, 794, 818
S			
\SavedUnexpanded	716, 764	\SOUL@everyosyllable	783, 795, 830
\SaveOrg	737, 754, 755, 756, 757, 758, 759, 760, 761, 762	\SOUL@everytoken	784, 833
\section	1018	\SOUL@exhyphen	238, 323, 459, 465
\setbox	863, 870, 884, 919	\SOUL@flushapo	190, 196, 209, 275, 281, 294, 507
\SetTestVBox	865, 872, 886, 918	\SOUL@flushcomma	188, 195, 208, 273, 280, 293, 490
\show	770, 773	\SOUL@flushgrave	192, 197, 210, 277, 282, 295, 524
\showbox	900, 901, 910, 911	\SOUL@flushminus	186, 194, 207, 271, 279, 292
\showboxbreadth	893	\SOUL@grave	191, 205, 276, 290, 525, 531, 533, 539
\showboxdepth	892	\SOUL@hyphkern	798, 805
\slash	235, 320, 1052	\SOUL@ignorespacesfalse	183, 268
\small	931, 971, 980	\SOUL@ignorespacestrue	230, 315
\SOUL@	786, 811, 839	\SOUL@minus	185, 199, 270, 284
\SOUL@@	181, 194, 195, 196, 197, 198, 200, 202, 204, 211, 216, 218, 226, 231, 235, 239, 241, 243, 249, 252, 255, 257, 266, 279, 280, 281, 282, 283, 285, 287, 289, 296, 301, 303, 311, 316, 320, 324, 326, 328, 334, 337, 340, 342, 386, 410	\SOUL@n	180, 212, 217, 225, 240, 242, 244, 256, 258, 260, 263, 265, 297, 302, 310, 325, 327, 329, 341, 343, 350, 353, 355, 357, 361
\SOuL@addfouroctets	357, 476	\SOuL@next	444, 446, 451, 460, 466, 468, 474
\SOUL@addgroup	256, 341, 541	\SOuL@octets	346, 349, 395, 422, 426, 430, 432
\SOUL@addmath	258, 343, 566	\SOuL@orgDeclareOption	117, 122
\SOuL@addprotect	240, 242, 325, 327, 591	\SOuL@orgDeclareRobustCommand	115, 120
\SOuL@addthreeoctets	355, 453	\SOuL@orgnewcommand	116, 121
\SOUL@addtoken	447, 448, 469, 470, 477, 478, 616	\SOuL@orgPackageError	118, 123
\SOuL@addtwooctets	353, 437	\SOuL@orgSOUL@error	129, 133
\SOuL@analyzeutfviii	345, 382	\SOUL@postamble	780, 821
\SOuL@analyzeutfviiix	347, 406	\SOUL@preamble	778, 790, 814
\SOUL@apo	189, 203, 274, 288, 508, 514, 516, 522	\SOuL@redefine	159, 179, 480, 490, 507, 524, 541, 566, 591, 616
\SOuL@AtEnd	81, 82, 635	\SOuL@remainingtoken	485, 487
\SOuL@charhash	378, 411, 416	\SOuL@restorelatexcmds	119, 126
\SOUL@charkern	808	\SOUL@scan	180, 217, 225, 247, 265, 302, 310, 332, 444, 460, 466, 552, 564, 577, 589, 602, 614, 624, 633
\SOuL@charthree	380, 425	\SOUL@sethyphenchar	799
\SOuL@chartwo	379, 429	\SOUL@setkern	798, 805, 808
\SOuL@checkutfviii	385, 390	\SOUL@setup	789
\SOuL@checkutfviiix	409, 415	\SOUL@spaceskip	246, 331
\SOuL@cmd	161, 166, 168, 177	\SOUL@spc	181, 218, 266, 303
\SOuL@cmdcheck	162, 165	\SOUL@splittoken	480
\SOUL@comma	187, 201, 272, 286, 491, 497, 499, 505	\SOUL@stop	211, 296, 480, 487
\SOuL@defsanitizedstring	363, 367, 368, 369, 370, 371, 372, 373, 374	\SOuL@stringcsnameu	412, 417
\SOUL@dotoken	260, 350	\SOuL@stringemdash	376, 462
\SOUL@doword	213, 219, 227, 232, 236, 245, 250, 253, 298, 304, 312, 317, 321, 330, 335, 338, 441, 457, 463	\SOuL@stringendash	375, 456
\SOuL@empty	381, 421	\SOuL@stringfour	400
\SOUL@error	128	\SOuL@stringmacrocolon	411, 416
\SOUL@eval	179	\SOuL@stringnobreakspace	377, 440
\SOUL@eventuallyexhyphen	214, 220, 228, 233, 237, 299, 305, 313, 318, 322, 442, 458, 464	\SOuL@stringoctets	387, 391
\SOUL@everyexhyphen	782, 804, 827	\SOuL@stringthree	398
\SOUL@everyhyphen	781, 796, 824	\SOuL@stringtwo	396
		\SOuL@stringundeferr	412, 417

