

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>	9
2.7	Actions for UTF-8 sequences	10
2.7.1	Redefinition of <code>\SOUL@splittoken</code>	11
2.8	Patches	11
3	Test	14
3.1	Catcode checks for loading	14
3.2	L ^A T _E Xtests	15
3.2.1	Driver definitions	16
3.2.2	Test macros	17
3.2.3	Testing	20
4	Installation	21
4.1	Download	21
4.2	Bundle installation	21
4.3	Package installation	22
4.4	Refresh file name databases	22
4.5	Some details for the interested	22
5	References	23

6 History	23
[2007/09/09 v1.0]	23
7 Index	23

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     \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 \endgroup
27 \expandafter\endinput
28 \fi
29 \endgroup

```

Package identification:

```

30 \begingroup
31 \catcode40 12 % (
32 \catcode41 12 % )
33 \catcode44 12 % ,
34 \catcode45 12 % -
35 \catcode46 12 % .
36 \catcode47 12 % /
37 \catcode58 12 % :
38 \catcode64 11 % @
39 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
40     \def\x#1#2#3[#4]{\endgroup
41         \immediate\write-1{Package: #3 #4}%
42         \xdef#1{#4}%
43     }%
44 \else
45     \def\x#1#2[#3]{\endgroup
46         #2[#{#3}]%
47         \ifx#1\relax
48             \xdef#1{#3}%
49         \fi
50     }%
51 \fi
52 \expandafter\x\csname ver@soulutf8.sty\endcsname
53 \ProvidesPackage{soulutf8}%
54 [2007/09/09 v1.0 Adding support for UTF-8 to soul (H0)]

```

2.2 Catcodes

```

55 \expandafter\edef\csname SOuL@AtEnd\endcsname{%
56     \catcode64 \the\catcode64\relax
57 }
58 \catcode64 11 % @
59 \def\TMP@EnsureCode#1#2{%
60     \edef\SOuL@AtEnd{%
61         \SOuL@AtEnd
62         \catcode#1 \the\catcode#1\relax
63     }%
64     \catcode#1 #2\relax
65 }
66 \TMP@EnsureCode{10}{12}% ^^J
67 \TMP@EnsureCode{33}{12}% !
68 \TMP@EnsureCode{34}{12}% "
69 \TMP@EnsureCode{36}{3}% $

```

```

70 \TMP@EnsureCode{39}{12}% '
71 \TMP@EnsureCode{40}{12}% (
72 \TMP@EnsureCode{41}{12}% )
73 \TMP@EnsureCode{42}{12}% *
74 \TMP@EnsureCode{43}{12}% +
75 \TMP@EnsureCode{44}{12}% ,
76 \TMP@EnsureCode{45}{12}% -
77 \TMP@EnsureCode{46}{12}% .
78 \TMP@EnsureCode{47}{12}% /
79 \TMP@EnsureCode{58}{12}% :
80 \TMP@EnsureCode{60}{12}% <
81 \TMP@EnsureCode{61}{12}% =
82 \TMP@EnsureCode{62}{12}% >
83 \TMP@EnsureCode{94}{7}% ^
84 \TMP@EnsureCode{96}{12}% '
85 \TMP@EnsureCode{126}{12}% \active % ~
86 \TMP@EnsureCode{128}{12}% ^^80
87 \TMP@EnsureCode{147}{12}% ^^93
88 \TMP@EnsureCode{148}{12}% ^^94
89 \TMP@EnsureCode{160}{12}% ^^a0
90 \TMP@EnsureCode{194}{12}% ^^c2
91 \TMP@EnsureCode{226}{12}% ^^e2

```

2.3 Loading packages

Package soul uses \documentclass to detect L^AT_EX.

```
92 \ifx\documentclass\@undefined
```

2.3.1 plain-T_EX

First we check, whether package soul is already loaded.

```
93 \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.

```

94 \let\Soul@orgDeclareRobustCommand\DeclareRobustCommand
95 \let\Soul@orgnewcommand\newcommand
96 \let\Soul@orgDeclareOption\DeclareOption
97 \let\Soul@orgPackageError\PackageError
98 \def\Soul@restorelatexcmds{%
99 \let\DeclareRobustCommand\Soul@orgDeclareRobustCommand
100 \let\newcommand\Soul@orgnewcommand
101 \let\DeclareOption\Soul@orgDeclareOption
102 \let\PackageError\Soul@orgPackageError
103 }%
104 \input soul.sty\relax
105 \Soul@restorelatexcmds
106 \fi

```

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

```

107 \input infwarerr.sty\relax
108 \let\Soul@orgSOUL@error\SOUL@error
109 \def\SOUL@error{%
110 \begingroup
111 \let\PackageError\@PackageError
112 \let\Soul@orgSOUL@error
113 \endgroup
114 }%
115 \input etexcmds.sty\relax

```

```

\@onelevel@sanitize Define LATEX's \@onelevel@sanitize if not already available.
116 \expandafter\ifx\csname @onelevel@sanitize\endcsname\relax
117 \def\@onelevel@sanitize#1{%
118 \edef#1{%
119 \expandafter\strip@prefix\meaning#1%
120 }%
121 }%

```

`\strip@prefix`

```

122 \def\strip@prefix#1>{%}%
123 \fi
124 \else

```

2.3.2 L^AT_EX

```

125 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{soul}}%
126 \ProcessOptions\relax
127 \RequirePackage{soul}[2003/11/17]%
128 \RequirePackage{infwarerr}[2007/09/09]%
129 \RequirePackage{etexcmds}[2007/09/09]%
130 \fi

```

2.3.3 ϵ -T_EX

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

```

131 \ifetex@unexpanded
132 \catcode33=14 % '!' : comment
133 \catcode43=9 % '+' : ignore
134 \else
135 \catcode33=9 % '!' : ignore
136 \catcode43=14 % '+' : comment
137 \fi

```

2.4 Macro for redefinitions

`\SOuL@redefine`

```

138 \def\SOuL@redefine#1{%
139 \begingroup
140 \def\SOuL@cmd{#1}%
141 \afterassignment\SOuL@cmdcheck
142 \def\SOuL@temp
143 }

```

`\SOuL@cmdcheck`

```

144 \def\SOuL@cmdcheck{%
145 \expandafter\ifx\SOuL@cmd\SOuL@temp
146 \else
147 \edef\SOuL@temp*{\expandafter\string\SOuL@cmd}%
148 \@PackageWarningNoLine{soulutf8}{%
149 Command \SOuL@temp* has changed.\MessageBreak
150 Supported versions of package 'soul': 2003/11/17.\MessageBreak
151 Depending on the unknown changes the redefinition\MessageBreak
152 of \SOuL@temp* may not behave correctly%
153 }%
154 \fi
155 \expandafter\endgroup
156 \expandafter\def\SOuL@cmd
157 }

```

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.

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

First the expected definition.

```
159 \def\SOUL@n*##1{\SOUL@scan}%
160 \if\noexpand\SOUL@@\SOUL@spc
161 \else
162 \SOUL@ignorespacesfalse
163 \fi
164 \ifnum\SOUL@minus=\thr@@
165 \SOUL@flushminus
166 \else\ifnum\SOUL@comma=\tw@
167 \SOUL@flushcomma
168 \else\ifnum\SOUL@apo=\tw@
169 \SOUL@flushapo
170 \else\ifnum\SOUL@grave=\tw@
171 \SOUL@flushgrave
172 \fi\fi\fi\fi
173 \ifx\SOUL@@-\else\SOUL@flushminus\fi
174 \ifx\SOUL@,@\else\SOUL@flushcomma\fi
175 \ifx\SOUL@@'\else\SOUL@flushapo\fi
176 \ifx\SOUL@@'\else\SOUL@flushgrave\fi
177 \ifx\SOUL@@-%
178 \advance\SOUL@minus\@ne
179 \else\ifx\SOUL@@,%
180 \advance\SOUL@comma\@ne
181 \else\ifx\SOUL@@'\%
182 \advance\SOUL@apo\@ne
183 \else\ifx\SOUL@@'\%
184 \advance\SOUL@grave\@ne
185 \else
186 \SOUL@flushminus
187 \SOUL@flushcomma
188 \SOUL@flushapo
189 \SOUL@flushgrave
190 \ifx\SOUL@@\SOUL@stop
191 \def\SOUL@n*{%
192 \SOUL@doword
193 \SOUL@eventuallyexhyphen\null
194 }%
195 \else\ifx\SOUL@@\par
196 \def\SOUL@n*\par{\par\leavevmode\SOUL@scan}%
197 \else\if\noexpand\SOUL@@\SOUL@spc
198 \SOUL@doword
199 \SOUL@eventuallyexhyphen\null
200 \ifSOUL@ignorespaces
201 \else
202 \SOUL@everyspace{}%
203 \fi
204 \def\SOUL@n* {\SOUL@scan}%
205 \else\ifx\SOUL@@\%
206 \SOUL@doword
207 \SOUL@eventuallyexhyphen\null
208 \SOUL@everyspace{\unskip\nobreak\hfil\break}%
209 \SOUL@ignorespacestrue
210 \else\ifx\SOUL@@~%
211 \SOUL@doword
212 \SOUL@eventuallyexhyphen\null
213 \SOUL@everyspace{\nobreak}%
```

[illegible]

Now the redefined version follows.

```

244 \def\SOUL@n##1{\SOUL@scan}%
245 \if\noexpand\SOUL@@\SOUL@spc
246 \else
247 \SOUL@ignorespacesfalse
248 \fi
249 \ifnum\SOUL@minus=\thr@@
250 \SOUL@flushminus
251 \else\ifnum\SOUL@comma=\tw@
252 \SOUL@flushcomma
253 \else\ifnum\SOUL@apo=\tw@
254 \SOUL@flushapo
255 \else\ifnum\SOUL@grave=\tw@
256 \SOUL@flushgrave
257 \fi\fi\fi\fi
258 \ifx\SOUL@@-\else\SOUL@flushminus\fi
259 \ifx\SOUL@@,\else\SOUL@flushcomma\fi
260 \ifx\SOUL@@'\else\SOUL@flushapo\fi
261 \ifx\SOUL@@\else\SOUL@flushgrave\fi
262 \ifx\SOUL@@-%
263 \advance\SOUL@minus\@ne
264 \else\ifx\SOUL@@,%
265 \advance\SOUL@comma\@ne
266 \else\ifx\SOUL@@'
267 \advance\SOUL@apo\@ne
268 \else\ifx\SOUL@@\%
269 \advance\SOUL@grave\@ne
270 \else
271 \SOUL@flushminus
272 \SOUL@flushcomma
273 \SOUL@flushapo
274 \SOUL@flushgrave

```

```

275 \ifx\SOUL@@\SOUL@stop
276 \def\SOUL@n*{%
277 \SOUL@doword
278 \SOUL@eventuallyexhyphen\null
279 }%
280 \else\ifx\SOUL@@\par
281 \def\SOUL@n*\par{\par\leavevmode\SOUL@scan}%
282 \else\if\noexpand\SOUL@@\SOUL@spc
283 \SOUL@doword
284 \SOUL@eventuallyexhyphen\null
285 \ifSOUL@ignorespaces
286 \else
287 \SOUL@everySPACE{}%
288 \fi
289 \def\SOUL@n* {\SOUL@scan}%
290 \else\ifx\SOUL@@\%
291 \SOUL@doword
292 \SOUL@eventuallyexhyphen\null
293 \SOUL@everySPACE{\unskip\nobreak\hfil\break}%
294 \SOUL@ignoreSPACEtrue
295 \else\ifx\SOUL@@~%
296 \SOUL@doword
297 \SOUL@eventuallyexhyphen\null
298 \SOUL@everySPACE{\nobreak}%
299 \else\ifx\SOUL@@\slash
300 \SOUL@doword
301 \SOUL@eventuallyexhyphen{/}%
302 \SOUL@exhyphen{/}%
303 \else\ifx\SOUL@@\mbox
304 \def\SOUL@n*{\SOUL@addprotect}%
305 \else\ifx\SOUL@@\hbox
306 \def\SOUL@n*{\SOUL@addprotect}%
307 \else\ifx\SOUL@@\soulomit
308 \def\SOUL@n*\soulomit##1{%
309 \SOUL@doword
310 {\spaceskip\SOUL@spaceskip##1}%
311 \SOUL@scan
312 }%
313 \else\ifx\SOUL@@\break
314 \SOUL@doword
315 \break
316 \else\ifx\SOUL@@\linebreak
317 \SOUL@doword
318 \SOUL@everySPACE{\linebreak}%
319 \else\ifcat\bgroup\noexpand\SOUL@@
320 \def\SOUL@n*{\SOUL@addgroup{}}%
321 \else\ifcat$\noexpand\SOUL@@
322 \def\SOUL@n*{\SOUL@addmath}%
323 \else

```

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

```

324 \SOUL@analyzeutfviii
325 \ifcase\SOUL@octets
326 \SOUL@analyzeutfviiiix
327 \fi
328 \ifcase\SOUL@octets
329 \def\SOUL@n*{\SOUL@dotoken}%
330 \or % 1
331 \or % 2
332 \def\SOUL@n*{\SOUL@addtwooctets}%
333 \or % 3
334 \def\SOUL@n*{\SOUL@addthreeoctets}%
335 \or % 4

```



```

336         \def\SOUL@n*{\SOUL@addfouroctets}%
337         \fi
338         \fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi
339         \fi\fi\fi\fi
340         \SOUL@n*%
341     }

```

2.6 UTF-8 analysis

2.6.1 Help strings

```

342 \def\SOUL@defsanitizedstring#1#2{%
343     \expandafter\def\csname SOUL@string#1\endcsname{#2}%
344     \expandafter\@onelevel@sanitize\csname SOUL@string#1\endcsname
345 }
346 \SOUL@defsanitizedstring{UTFviii}{UTFviii@}
347 \SOUL@defsanitizedstring{octets}{@octets}
348 \SOUL@defsanitizedstring{two}{two}
349 \SOUL@defsanitizedstring{three}{three}
350 \SOUL@defsanitizedstring{four}{four}
351 \SOUL@defsanitizedstring{macrocolon}{macro:}
352 \SOUL@defsanitizedstring{csnameu}{csname u8-}
353 \SOUL@defsanitizedstring{undeferr}{utf@viii@undeferr}
354 \def\SOUL@stringendash{^e2^80^93}
355 \def\SOUL@stringemdash{^e2^80^94}
356 \def\SOUL@stringnobreakspace{^c2^a0}
357 \edef\SOUL@charhash{\string #}
358 \edef\SOUL@chartwo{\string 2}
359 \edef\SOUL@charthree{\string 3}
360 \def\SOUL@empty{}

```

2.6.2 Support for utf8.def

\SOUL@analyzeutfviii

```

361 \begingroup
362     \edef\x{\endgroup
363         \def\noexpand\SOUL@analyzeutfviii{%
364             \noexpand\expandafter\noexpand\SOUL@checkutfviii
365             \noexpand\meaning\noexpand\SOUL@@
366             \SOUL@stringUTFviii\SOUL@stringoctets
367             \noexpand\@nil
368         }%
369         \def\noexpand\SOUL@checkutfviii
370             ##1\SOUL@stringUTFviii##2\SOUL@stringoctets##3\noexpand\@nil
371     }%
372 \x{%
373     \def\SOUL@temp{#2}%
374     \chardef\SOUL@octets=%
375         \ifx\SOUL@temp\SOUL@stringtwo
376             \tw@
377         \else\ifx\SOUL@temp\SOUL@stringthree
378             \thr@@
379         \else\ifx\SOUL@temp\SOUL@stringfour
380             4 %
381         \else
382             \z@
383         \fi\fi\fi
384 }

```

2.6.3 Support for utf8x.def

\SOUL@analyzeutfviii

```

385 \begingroup

```

```

386 \edef\x{\endgroup
387 \def\noexpand\Soul@analyzeutfviiiix{%
388 \noexpand\expandafter\noexpand\Soul@checkutfviiiix
389 \noexpand\meaning\noexpand\Soul@@
390 \Soul@stringmacrocolon\Soul@charhash1{}{}{}{}%
391 \Soul@stringcsnameu\Soul@stringundeferr
392 \noexpand\@nil
393 }%

\Soul@checkutfviiiix

394 \def\noexpand\Soul@checkutfviiiix
395 ##1\Soul@stringmacrocolon\Soul@charhash1##2##3##4##5##6%
396 \Soul@stringcsnameu##7\Soul@stringundeferr##8\noexpand\@nil
397 }%
398 \x{%
399 \def\Soul@temp{#7}%
400 \ifx\Soul@temp\Soul@empty
401 \chardef\Soul@octets=\z@
402 \else
403 \def\Soul@temp{#5}%
404 \ifx\Soul@temp\Soul@charthree
405 \chardef\Soul@octets=4 %
406 \else
407 \def\Soul@temp{#3}%
408 \ifx\Soul@temp\Soul@chartwo
409 \chardef\Soul@octets=\thr@@
410 \else
411 \chardef\Soul@octets=\tw@
412 \fi
413 \fi
414 \fi
415 }

```

2.7 Actions for UTF-8 sequences

\Soul@addtwooctets

```

416 \def\Soul@addtwooctets#1#2{%
417 \def\Soul@temp{#1#2}%
418 \@onelevel@sanitize\Soul@temp
419 \ifx\Soul@temp\Soul@stringnobreakspace
420 \Soul@doword
421 \Soul@eventuallyexhyphen\null
422 \Soul@everySPACE{\nobreak}%
423 \let\Soul@next\Soul@scan
424 \else
425 \def\Soul@next{%
426 ! \Soul@addtoken{\noexpand#1\noexpand#2}}%
427 + \Soul@addtoken{\etex@unexpanded{#1#2}}}%
428 }%
429 \fi
430 \Soul@next
431 }

```

\Soul@addthreeoctets

```

432 \def\Soul@addthreeoctets#1#2#3{%
433 \def\Soul@temp{#1#2#3}%
434 \@onelevel@sanitize\Soul@temp
435 \ifx\Soul@temp\Soul@stringendash
436 \Soul@doword
437 \Soul@eventuallyexhyphen{-}%
438 \Soul@exhyphen{--}%
439 \let\Soul@next\Soul@scan

```

```

440 \else
441   \ifx\Soul@temp\Soul@stringemdash
442     \SOUL@doword
443     \SOUL@eventuallyexhyphen{-}%
444     \SOUL@exhyphen{---}%
445     \let\Soul@next\Soul@scan
446   \else
447     \def\Soul@next{%
448       \SOUL@addtoken{\noexpand#1\noexpand#2\noexpand#3}%
449 +     \SOUL@addtoken{\etex@unexpanded{#1#2#3}}}%
450     }%
451   \fi
452 \fi
453 \Soul@next
454 }

```

`\Soul@addfouroctets`

```

455 \def\Soul@addfouroctets#1#2#3#4{%
456 ! \SOUL@addtoken{\noexpand#1\noexpand#2\noexpand#3\noexpand#4}%
457 + \SOUL@addtoken{\etex@unexpanded{#1#2#3#4}}}%
458 }

```

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.

```

459 \Soul@redefine\Soul@splittoken#1#2\Soul@stop{%
460   \global\Soul@token={#1}%
461   \global\Soul@word={#2}%
462 }#1{%
463   \global\Soul@token={#1}%
464   \Soul@remainingtoken\relax
465 }

```

`\Soul@remainingtoken`

```

466 \def\Soul@remainingtoken#1\Soul@stop{%
467   \global\Soul@word=\expandafter{\@gobble#1}%
468 }

```

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`

```

469 \Soul@redefine\Soul@flushcomma{%
470   \ifcase\Soul@comma
471   \or
472     \edef\x{\Soul@word={\the\Soul@word,}}\x
473   \or
474     \edef\x{\Soul@word={\the\Soul@word{,}}}\x
475   \fi
476   \SOUL@comma\z@
477 }{%
478   \ifcase\Soul@comma
479   \or

```

```

480 \edef\x{\SOUL@word={\the\SOUL@word,}}\x
481 \or
482 \edef\x{\SOUL@word={\the\SOUL@word{,,}}\x
483 \fi
484 \SOUL@comma\z@
485 }

```

\SOUL@flushapo

```

486 \SOUL@redefine\SOUL@flushapo{%
487 \ifcase\SOUL@apo
488 \or
489 \edef\x{\SOUL@word={\the\SOUL@word'}\x
490 \or
491 \edef\x{\SOUL@word={\the\SOUL@word{' '}}\x
492 \fi
493 \SOUL@apo\z@
494 }{%
495 \ifcase\SOUL@apo
496 \or
497 \edef\x{\SOUL@word={\the\SOUL@word'}\x
498 \or
499 \edef\x{\SOUL@word={\the\SOUL@word{' '}}\x
500 \fi
501 \SOUL@apo\z@
502 }

```

\SOUL@flushgrave

```

503 \SOUL@redefine\SOUL@flushgrave{%
504 \ifcase\SOUL@grave
505 \or
506 \edef\x{\SOUL@word={\the\SOUL@word'}\x
507 \or
508 \edef\x{\SOUL@word={\the\SOUL@word{' '}}\x
509 \fi
510 \SOUL@grave\z@
511 }{%
512 \ifcase\SOUL@grave
513 \or
514 \edef\x{\SOUL@word={\the\SOUL@word'}\x
515 \or
516 \edef\x{\SOUL@word={\the\SOUL@word{' '}}\x
517 \fi
518 \SOUL@grave\z@
519 }

```

\SOUL@addgroup

```

520 \SOUL@redefine\SOUL@addgroup#1#2{%
521 {%
522 \let\protect\noexpand
523 \edef\x{%
524 \global\SOUL@word={%
525 \the\SOUL@word
526 {\noexpand#1#2}}%
527 }%
528 }%
529 \x
530 }%
531 \SOUL@scan
532 }#1#2{%
533 \begingroup
534 \let\protect\noexpand
535 \edef\x{\endgroup

```

```

536      \SOUL@word={%
537      \the\SOUL@word
538 !      {\noexpand#1{#2}}}%
539 +      {\etex@unexpanded{#1{#2}}}%
540      }%
541      }%
542      \x
543      \SOUL@scan
544 }

```

\SOUL@addmath

```

545 \SOUL@redefine\SOUL@addmath$#1${%
546   {%
547     \let\protect\noexpand
548     \edef\x{%
549       \global\SOUL@word={%
550       \the\SOUL@word
551       {\hbox{ $#1$ }}}%
552     }%
553     }%
554     \x
555     }%
556     \SOUL@scan
557 }$#1${%
558   \begingroup
559   \let\protect\noexpand
560   \edef\x{\endgroup
561     \SOUL@word={%
562     \the\SOUL@word
563 !     {\hbox{ $#1$ }}}%
564 +     {\etex@unexpanded{\hbox{ $#1$ }}}%
565     }%
566     }%
567     \x
568     \SOUL@scan
569 }

```

\SOUL@addprotect

```

570 \SOUL@redefine\SOUL@addprotect#1#2{%
571   {%
572     \let\protect\noexpand
573     \edef\x{%
574       \global\SOUL@word={%
575       \the\SOUL@word
576       {\hbox{ #2 }}}%
577     }%
578     }%
579     \x
580     }%
581     \SOUL@scan
582 }#1#2{%
583   \begingroup
584   \let\protect\noexpand
585   \edef\x{\endgroup
586     \SOUL@word={%
587     \the\SOUL@word
588 !     {\hbox{ #2 }}}%
589 +     {\etex@unexpanded{\hbox{ #2 }}}%
590     }%
591     }%
592     \x
593     \SOUL@scan

```

```

594 }

\SOUL@addtoken

595 + \SOUL@redefine\SOUL@addtoken#1{%
596 +   \edef\x{%
597 +     \SOUL@word={%
598 +       \the\SOUL@word
599 +       \noexpand#1%
600 +     }%
601 +   }%
602 +   \x
603 +   \SOUL@scan
604 + }#1{%
605 +   \edef\x{%
606 +     \SOUL@word={%
607 +       \the\SOUL@word
608 +       \etex@unexpanded{#1}%
609 +     }%
610 +   }%
611 +   \x
612 +   \SOUL@scan
613 + }%

614 \SOUL@AtEnd
615 </package>

```

3 Test

3.1 Catcode checks for loading

```

616 <*test1>

617 \catcode'\@=11 %
618 \def\RestoreCatcodes{}
619 \count@=0 %
620 \loop
621   \edef\RestoreCatcodes{%
622     \RestoreCatcodes
623     \catcode\the\count@=\the\catcode\count@\relax
624   }%
625 \ifnum\count@<255 %
626   \advance\count@\@ne
627 \repeat
628
629 \def\RangeCatcodeInvalid#1#2{%
630   \count@=#1\relax
631   \loop
632     \catcode\count@=15 %
633     \ifnum\count@<#2\relax
634       \advance\count@\@ne
635     \repeat
636 }
637 \def\Test{%
638   \RangeCatcodeInvalid{0}{47}%
639   \RangeCatcodeInvalid{58}{64}%
640   \RangeCatcodeInvalid{91}{96}%
641   \RangeCatcodeInvalid{123}{255}%
642   \catcode'\@=12 %
643   \catcode'\=0 %
644   \catcode'\{=1 %
645   \catcode'\}=2 %
646   \catcode'\#=6 %

```

```

647 \catcode'\[=12 %
648 \catcode'\]=12 %
649 \catcode'\%=14 %
650 \catcode'\ =10 %
651 \catcode13=5 %
652 \input soulutf8.sty\relax
653 \RestoreCatcodes
654 }
655 \Test
656 \csname @@end\endcsname
657 \end
658 </test1>

```

3.2 L^AT_EX tests

```

659 <*test2 | test3 | test4 | test5>
660 <noetex> \let\unexpanded\UNDEFINED
661 \NeedsTeXFormat{LaTeX2e}
662 \documentclass{article}
663 \usepackage[T1]{fontenc}
664 \usepackage{lmodern}
665 \usepackage{textcomp}
666 \usepackage[ngerman]{babel}
667 \expandafter\def\csname ver@hypdestopt.sty\endcsname{}
668 \usepackage[unicode]{hyperref}
669 \usepackage{bookmark}
670 \usepackage{color}
671 \usepackage{soul}[2003/11/17]
672 \makeatletter
673 \newcommand*{\@ifDefined}[1]{%
674   \@ifundefined{#1}{%
675     \latex@error{%
676       \expandafter\noexpand\csname#1\endcsname undefined%
677     }\@ehc
678   }%
679 }
680 \newcommand*{\SaveOrg}[1]{%
681   \@ifdefinable{ORG@SOUL@#1}{%
682     \@ifDefined{SOUL@#1}{%
683       \expandafter\let\csname ORG@SOUL@#1\expandafter\endcsname
684         \csname SOUL@#1\endcsname
685     }%
686   }%
687 }
688 \newcommand*{\RestoreOrg}[1]{%
689   \@ifDefined{ORG@SOUL@#1}{%
690     \@ifDefined{SOUL@#1}{%
691       \expandafter\let\csname SOUL@#1\expandafter\endcsname
692         \csname ORG@SOUL@#1\endcsname
693     }%
694   }%
695 }
696 \makeatother
697 \SaveOrg{eval}
698 \SaveOrg{splittoken}
699 \SaveOrg{flushcomma}
700 \SaveOrg{flushapo}
701 \SaveOrg{grave}
702 \SaveOrg{addgroup}
703 \SaveOrg{addmath}
704 \SaveOrg{addprotect}
705 \SaveOrg{addtoken}
706 \usepackage{soulutf8}[2007/09/09]

```

Input encoding utf8.

```
707 <utf8>
708 \usepackage[utf8]{inputenc}
709 \DeclareUnicodeCharacter{00A0}{\nobreakspace}
710 </utf8>
```

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

```
711 <utf8x>
712 \let\origshow\show
713 \usepackage[utf8x]{inputenc}
714 \AtBeginDocument{%
715   \let\show\origshow
716 }
717 </utf8x>
```

3.2.1 Driver definitions

```
718 \makeatletter
```

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

```
719 \DeclareRobustCommand*\an{%
720   \def\Soul@preamble{$^{P}}%
721   \def\Soul@everyspace##1{##1\texttt{\char'\ }\\linebreak[2]}%
722   \def\Soul@postamble{$^{E}}%
723   \def\Soul@everyhyphen{$^{~}}$\linebreak[1]}%
724   \def\Soul@everyexhyphen##1{##1$^{~=}}%
725   \def\Soul@everysyllable{$^{S}}%
726   \def\Soul@everytoken{\the\Soul@token$^{T}}%
727   \def\Soul@everylowerthan{$^{L}}%
728   \Soul@
729 }
```

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

```
730 \DeclareRobustCommand*\sy{%
731   \Soul@setup
732   \def\Soul@preamble{%
733     \lefthyphenmin=\z@
734     \righthyphenmin=\z@
735   }%
736   \def\Soul@everyspace##1{##1\space}%
737   \def\Soul@everysyllable{\the\Soul@syllable}%
738   \def\Soul@everyhyphen{%
739     \discretionary{%
740       \Soul@setkern\Soul@hyphkern
741       \Soul@sethyphenchar
742     }{}{%
743       \hbox{\kern1pt$\cdot$}%
744     }%
745   }%
746   \def\Soul@everyexhyphen##1{%
747     \Soul@setkern\Soul@hyphkern
748     \hbox{##1}%
749     \discretionary{}{}{%
750       \Soul@setkern\Soul@charkern
751     }%
752   }%
753   \Soul@
754 }
```

\soulH

```
755 \DeclareRobustCommand*\soulH{%
756   \def\Soul@preamble{%
757     \gdef\SoulH{}%
758     \soulHadd{ }%
```



```

759 }%
760 \def\SOUL@everyspace##1{%
761   \soulHadd{Sp}%
762 }%
763 \def\SOUL@postamble{%
764   \soulHadd{]}%
765 }%
766 \def\SOUL@everyhyphen{%
767   \soulHadd{Hy}%
768 }%
769 \def\SOUL@everyexhyphen##1{%
770   \soulHadd{Exhy}%
771 }%
772 \def\SOUL@everysyllable{%
773   \soulHadd{Sy}%
774 }%
775 \def\SOUL@everytoken{%
776   \soulHadd{T}%
777 }%
778 \def\SOUL@everylowerthan{%
779   \soulHadd{<}%
780 }%
781 \SOUL@
782 }
783 \newcommand*{\soulHadd}[1]{%
784   #1%
785   \expandafter\gdef\expandafter\soulHstring\expandafter{%
786     \soulHstring
787     #1%
788   }%
789 }

```

3.2.2 Test macros

```

790 \usepackage{qstest}
791 \IncludeTests{*}
792 \LogTests{log}{*}{*}

793 \newsavebox\boxA
794 \newsavebox\boxB
795 \newsavebox\boxC

\TestDriver

796 \newcommand*{\TestDriver}[4]{%
797   \subsection{Driver #1}%
798   \@PackageInfo{soulutf8-test}{* Driver #1}%
799   \expandafter\let\expandafter\testdriver\csname #1\endcsname
800   \def\teststringA{#3}%
801   \def\teststringB{#4}%
802   \begin{qstest}{string-#2-driver-#1}{string,#2,driver,#1}%
803     \@PackageInfo{soulutf8-test}{* Version: soulutf8}%
804     \global\let\soulHstring\@empty
805     \setbox0=\hbox{\expandafter\testdriver\expandafter{\teststringA}}%
806     \let\HstringA\soulHstring
807     \SetTestVBox{\boxA}{\teststringA}%
808     %
809     \@PackageInfo{soulutf8-test}{* Version: soul + patch}%
810     \global\let\soulHstring\@empty
811     \RestoreOrg{eval}%
812     \setbox2=\hbox{\expandafter\testdriver\expandafter{\teststringB}}%
813     \let\HstringB\soulHstring
814     \SetTestVBox{\boxB}{\teststringB}%
815     %

```

```

816 \@PackageInfo{soulutf8-test}{* Version: original soul}%
817 \global\let\soulHstring\@empty
818 \RestoreOrg{splittoken}%
819 \RestoreOrg{flushcomma}%
820 \RestoreOrg{flushapo}%
821 \RestoreOrg{grave}%
822 \RestoreOrg{addgroup}%
823 \RestoreOrg{addmath}%
824 \RestoreOrg{addprotect}%
825 \RestoreOrg{addtoken}%
826 \setbox4=\hbox{\expandafter\testdriver\expandafter{\teststringB}}%
827 \let\HstringC\soulHstring
828 \SetTestVBox{\boxC}{\teststringB}%
829 %
830 \TestInfo{0}{soulutf8}%
831 \TestInfo{2}{soul + patch}%
832 \TestInfo{4}{original soul}%
833 \TestInfoVBox
834 \showboxdepth=\maxdimen
835 \showboxbreadth=\maxdimen
836 \tracingonline=\@ne
837 \Expect*{\HstringA}*{\HstringB}%
838 \Expect*{\HstringA}*{\HstringC}%
839 \Expect*{\the\wd0}*{\the\wd2}%
840 \ifdim\wd0=\wd2 %
841 \else
842 \showbox0 %
843 \showbox2 %
844 \fi
845 \Expect*{\the\ht0}*{\the\ht2}%
846 \Expect*{\the\dp0}*{\the\dp2}%
847 \Expect*{\the\wd\boxB}*{\the\wd\boxA}%
848 \Expect*{\the\wd\boxC}*{\the\wd\boxA}%
849 \Expect*{\the\ht\boxB}*{\the\ht\boxA}%
850 \ifdim\ht\boxB=\ht\boxA
851 \else
852 \showbox\boxA
853 \showbox\boxB
854 \fi
855 \Expect*{\the\ht\boxC}*{\the\ht\boxA}%
856 \Expect*{\the\dp\boxB}*{\the\dp\boxA}%
857 \Expect*{\the\dp\boxC}*{\the\dp\boxA}%
858 \end{qstest}%
859 }

```

\SetTestVBox

```

860 \newcommand*\SetTestVBox}[2]{%
861 \setbox#1=\vbox{%
862 \hsize=0pt\relax
863 \hfuzz=100mm\relax
864 \expandafter\testdriver\expandafter{#2}%
865 }%
866 }

```

\TestInfo

```

867 \newcommand*\TestInfo}[2]{%
868 \par
869 \noindent
870 \copy#1\relax
871 \hfill
872 \begin{group}
873 \small

```

```

874 \ifdim\wd0=\wd2\relax
875 \ifdim\wd0=\wd4\relax
876 \else
877 \ifnum#1=2 %
878 \raisebox{.5\baselineskip}[Opt][Opt]{%
879 \llap{$\}$ }%
880 \rlap{width = \the\wd#1\relax\space$\{ $\}%
881 \phantom{width = 999.99999pt}}%
882 \quad
883 }%
884 \else
885 \ifnum#1=4 %
886 \rlap{width = \the\wd#1}%
887 \phantom{width = 999.99999pt}}%
888 \quad
889 \fi
890 \fi
891 \fi
892 \else
893 \begin{group}
894 \bfseries
895 \rlap{width = \the\wd#1}%
896 \phantom{width = 999.99999pt}}%
897 \end{group}
898 \quad
899 \fi
900 \rlap{(#2)}%
901 \phantom{(original soul)}}%
902 \end{group}
903 \par
904 }

```

\TestInfoVBox

```

905 \newcommand*{\TestInfoVBox}{%
906 \par
907 \noindent
908 \mbox{\vrule\relax\space\copy\boxA}%
909 \ifdim\ht\boxA=\ht\boxB
910 \else
911 \hfill
912 \begin{group}
913 \small
914 \rlap{(soulutf8)}%
915 \phantom{(soulutf8)}}%
916 \end{group}
917 \par
918 \noindent
919 \mbox{\vrule\relax\space\copy\boxB}%
920 \hfill
921 \begin{group}
922 \small
923 \rlap{(soul)}%
924 \phantom{(soulutf8)}}%
925 \end{group}
926 \fi
927 \par
928 }

```

\Test

```

929 \makeatletter
930 \newcommand*{\Test}[3]{%
931 \clearpage

```

```

932 \begin{group
933   \def\linebreak{\textbackslash linebreak }%
934   \def\{\{\textbackslash\textbackslash}%
935   \def\soulomit##1{%
936     \textbackslash soulomit\textbraceleft##1\textbraceright
937   }%
938   \def\mbox##1{%
939     \textbackslash mbox\textbraceleft##1\textbraceright
940   }%
941   \def\abc{\textbackslash abc }%
942   \def\texttt##1{%
943     \textbackslash texttt\textbraceleft##1\textbraceright
944   }%
945   \def\c##1{%
946     \textbackslash c\textbraceleft##1\textbraceright
947   }%
948 <*utf8x>
949   \count@=128 %
950   \loop
951     \ifnum\count@<256 %
952       \lccode'\~=\count@
953       \lowercase{%
954         \def~{\noexpand~}%
955       }%
956       \advance\count@\@ne
957     \repeat
958 </utf8x>
959   \csname protected@edef\endcsname\x{\endgroup
960     \noexpand\section{String #1: #2}%
961   }%
962   \x
963   \TestDriver{so}{#1}{#2}{#3}%
964   \TestDriver{caps}{#1}{#2}{#3}%
965   \TestDriver{ul}{#1}{#2}{#3}%
966   \TestDriver{st}{#1}{#2}{#3}%
967   \TestDriver{hl}{#1}{#2}{#3}%
968   \TestDriver{an}{#1}{#2}{#3}%
969   \TestDriver{sy}{#1}{#2}{#3}%
970   \TestDriver{soulH}{#1}{#2}{#3}%
971 }
972 \makeatother

```

3.2.3 Testing

```

973 \begin{document}
974 \Test A{}{}%
975 \Test B{Hello World}{Hello World}%
976 \Test C{%
977   Umlauts: ^^c3^^84^^c3^^96^^c3^^9c^^c3^^a4^^c3^^b6^^c3^^bc^^c3^^9f%
978 }{%
979   Umlauts: \"A\"O\"U\"a\"o\"u\\ss
980 }%
981 \Test D{Euro: ^^e2^^82^^ac}{Euro: \texteuro}%
982 \Test E{Stra^^c3^^9fe}{Stra\\ss e}%
983 \Test F{R^^c3^^b6mer}{R\"omer}%
984 \Test G{A^^e2^^80^^93B^^e2^^80^^94C}{A--B---C}%
985 \Test H{X^^c2^^a0Y}{X~Y}%
986 \Test I{A^^c5^^a4}{A\\v T}%
987 \Test J{A,B,,C,,,D,,,E}{A,B,,C,,,D,,,E}%
988 \Test K{A'B''C''''D''''E}{A'B''C''''D''''E}%
989 \Test L{A'B'C'D'E}{A'B'C'D'E}%
990 \Test M{A$x$B}{A$x$B}%
991 \Test N{A\mbox{abc}B}{A\mbox{abc}B}%

```

```

992 \Test O{,TA''}{,TA''}%
993 \Test P{{!'}HOLA{?'}}{{!'}HOLA{?'}}%
994 \Test Q{re-sent\slash x}{re-sent\slash x}%
995 \Test R{new\\line}{new\\line}%
996 \Test S{foo\linebreak bar}{foo\linebreak bar}%
997 \Test T{Virchow{sche}}{Virchow{sche}}%
998 \Test U{A\soulomit{foo}B}{A\soulomit{foo}B}%
999 \Test V{A\texttt{xyz}B}{A\texttt{xyz}B}%
1000 \Test W{ffi}{ffi}%
1001 \Test X{f{}f{}i}{f{}f{}i}%
1002 \Test Y{f\mbox{}f\mbox{}i}{f\mbox{}f\mbox{}i}%
1003 \Test Z{c{"a}}{c{"a}}%
1004 <*test2 | test3>
1005 \let\abcx\UNDEFINED
1006 \def\abc{%
1007     \begingroup
1008         \def\abcx{x}%
1009         \abcx
1010     \endgroup
1011 }%
1012 \Test{AA}{[{x\abc x}]}[{xxx}]}%
1013 </test2 | test3>
1014 \end{document}
1015 </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:macros/latex/contrib/oberdiek/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
```

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/utf8inc.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>	979, 983, 1003
<code>\#</code>	646
<code>\%</code>	649
<code>\@</code>	617, 642
<code>\@PackageError</code>	111
<code>\@PackageInfo</code>	798, 803, 809, 816
<code>\@PackageWarningNoLine</code>	148
<code>\@ehc</code>	677
<code>\@empty</code>	804, 810, 817
<code>\@gobble</code>	467
<code>\@ifDefined</code>	673, 682, 689, 690
<code>\@ifdefinable</code>	681
<code>\@ifundefined</code>	674
<code>\@latex@error</code>	675
<code>\@one</code>	178, 180, 182, 184, 263, 265, 267, 269, 626, 634, 836, 956
<code>\@nil</code>	367, 370, 392, 396
<code>\@onelevel@sanitize</code>	<u>116</u> , 344, 418, 434
<code>\@undefined</code>	92
<code>\[</code>	647
<code>\]</code>	205, 290, 643, 934, 995
<code>\{</code>	644, 880
<code>\}</code>	645, 879
<code>\]</code>	648
<code>\~</code>	952
<code>_</code>	650, 721
A	
<code>\abc</code>	941, 1006, 1012
<code>\abcx</code>	1005, 1008, 1009
<code>\active</code>	85
<code>\advance</code>	178, 180, 182, 184, 263, 265, 267, 269, 626, 634, 956
<code>\afterassignment</code>	141
B	
<code>\baselineskip</code>	878
<code>\begin</code>	802, 973
<code>\bfseries</code>	894
<code>\boxA</code>	793, 807, 847, 848, 849, 850, 852, 855, 856, 857, 908, 909
<code>\boxB</code>	794, 814, 847, 849, 850, 853, 856, 909, 919
<code>\boxC</code>	795, 828, 848, 855, 857
<code>\break</code>	208, 228, 230, 293, 313, 315
C	
<code>\c</code>	945, 1003
<code>\catcode</code>	3, 4, 5, 6, 7, 31, 32, 33, 34, 35, 36, 37, 38, 56, 58, 62, 64, 132, 133, 135, 136, 617, 623, 632, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651
<code>\cdot</code>	743
<code>\char</code>	721
<code>\chardef</code>	374, 401, 405, 409, 411
<code>\clearpage</code>	931
<code>\copy</code>	870, 908, 919
<code>\count@</code>	619, 623, 625, 626, 630, 632, 633, 634, 949, 951, 952, 956
<code>\csname</code>	8, 18, 39, 52, 55, 93, 116, 343, 344, 656, 667, 676, 683, 684, 691, 692, 799, 959
<code>\CurrentOption</code>	125
D	
<code>\DeclareOption</code>	96, 101, 125
<code>\DeclareRobustCommand</code>	94, 99, 719, 730, 755

<code>\DeclareUnicodeCharacter</code>	709	<code>\llap</code>	879
<code>\discretionary</code>	739, 749	<code>\LogTests</code>	792
<code>\documentclass</code>	92, 662	<code>\loop</code>	620, 631, 950
<code>\dp</code>	846, 856, 857	<code>\lowercase</code>	953
E		M	
<code>\empty</code>	12	<code>\makeatletter</code>	672, 718, 929
<code>\end</code>	657, 858, 1014	<code>\makeatother</code>	696, 972
<code>\endcsname</code>	8, 18, 39, 52, 55, 93, 116, 343, 344, 656, 667, 676, 683, 684, 691, 692, 799, 959	<code>\maxdimen</code>	834, 835
<code>\endinginput</code>	27	<code>\mbox</code>	218, 303, 908, 919, 938, 991, 1002
<code>\etex@unexpanded</code>	427, 449, 457, 539, 564, 589, 608	<code>\meaning</code>	119, 365, 389
<code>\Expect</code>	837, 838, 839, 845, 846, 847, 848, 849, 855, 856, 857	<code>\MessageBreak</code>	149, 150, 151
G		N	
<code>\gdef</code>	757, 785	<code>\NeedsTeXFormat</code>	661
H		<code>\newcommand</code>	95, 100, 673, 680, 688, 783, 796, 860, 867, 905, 930
<code>\hbox</code>	220, 305, 551, 563, 564, 576, 588, 589, 743, 748, 805, 812, 826	<code>\newsavebox</code>	793, 794, 795
<code>\hfil</code>	208, 293	<code>\nobreak</code>	208, 213, 293, 298, 422
<code>\hfill</code>	871, 911, 920	<code>\nobreakspace</code>	709
<code>\hfuzz</code>	863	<code>\noindent</code>	869, 907, 918
<code>\hsize</code>	862	<code>\null</code>	193, 199, 207, 212, 278, 284, 292, 297, 421
<code>\HstringA</code>	806, 837, 838	O	
<code>\HstringB</code>	813, 837	<code>\origshow</code>	712, 715
<code>\HstringC</code>	827, 838	P	
<code>\ht</code>	845, 849, 850, 855, 909	<code>\PackageError</code>	97, 102, 111
I		<code>\PackageInfo</code>	23
<code>\if</code>	160, 197, 245, 282	<code>\par</code>	195, 196, 280, 281, 868, 903, 906, 917, 927
<code>\ifcase</code>	9, 325, 328, 470, 478, 487, 495, 504, 512	<code>\PassOptionsToPackage</code>	125
<code>\ifcat</code>	234, 236, 319, 321	<code>\phantom</code>	881, 887, 896, 901, 915, 924
<code>\ifdim</code>	840, 850, 874, 875, 909	<code>\ProcessOptions</code>	126
<code>\ifetex@unexpanded</code>	131	<code>\protect</code>	522, 534, 547, 559, 572, 584
<code>\ifnum</code>	164, 166, 168, 170, 249, 251, 253, 255, 625, 633, 877, 885, 951	<code>\ProvidesPackage</code>	53
<code>\ifSOUL@ignorespaces</code>	200, 285	Q	
<code>\ifx</code>	10, 12, 18, 39, 47, 92, 93, 116, 145, 173, 174, 175, 176, 177, 179, 181, 183, 190, 195, 205, 210, 214, 218, 220, 222, 228, 231, 258, 259, 260, 261, 262, 264, 266, 268, 275, 280, 290, 295, 299, 303, 305, 307, 313, 316, 375, 377, 379, 400, 404, 408, 419, 435, 441	<code>\qqquad</code>	882, 888, 898
<code>\immediate</code>	20, 41	R	
<code>\IncludeTests</code>	791	<code>\raisebox</code>	878
<code>\input</code>	104, 107, 115, 652	<code>\RangeCatcodeInvalid</code>	629, 638, 639, 640, 641
K		<code>\repeat</code>	627, 635, 957
<code>\kern</code>	743	<code>\RequirePackage</code>	127, 128, 129
L		<code>\RestoreCatcodes</code>	618, 621, 622, 653
<code>\lccode</code>	952	<code>\RestoreOrg</code>	688, 811, 818, 819, 820, 821, 822, 823, 824, 825
<code>\leavevmode</code>	196, 281	<code>\righthyphenmin</code>	734
<code>\lefthyphenmin</code>	733	<code>\rlap</code>	880, 886, 895, 900, 914, 923
<code>\linebreak</code>	231, 233, 316, 318, 721, 723, 933, 996	S	
M		<code>\SaveOrg</code>	680, 697, 698, 699, 700, 701, 702, 703, 704, 705
N		<code>\section</code>	960
O		<code>\setbox</code>	805, 812, 826, 861
P		<code>\SetTestVBox</code>	807, 814, 828, 860
Q		<code>\show</code>	712, 715
R		<code>\showbox</code>	842, 843, 852, 853
S		<code>\showboxbreadth</code>	835
		<code>\showboxdepth</code>	834
		<code>\slash</code>	214, 299, 994
		<code>\small</code>	873, 913, 922

\SOUL@	728, 753, 781	\SOUL@grave	170,
\SOUL@@	160, 173, 174,	184, 255, 269, 504, 510, 512, 518	
175, 176, 177, 179, 181, 183,		\SOUL@hyphkern	740, 747
190, 195, 197, 205, 210, 214,		\SOUL@ignorespacesfalse	162, 247
218, 220, 222, 228, 231, 234,		\SOUL@ignorespacestrue	209, 294
236, 245, 258, 259, 260, 261,		\SOUL@minus	164, 178, 249, 263
262, 264, 266, 268, 275, 280,		\SOUL@n	159, 191, 196, 204, 219, 221,
282, 290, 295, 299, 303, 305,		223, 235, 237, 239, 242, 244,	
307, 313, 316, 319, 321, 365, 389		276, 281, 289, 304, 306, 308,	
\SOuL@addfouroctets	336, 455	320, 322, 329, 332, 334, 336, 340	
\SOUL@addgroup	235, 320, 520	\SOuL@next	
\SOUL@addmath	237, 322, 545	. 423, 425, 430, 439, 445, 447, 453	
\SOUL@addprotect	219, 221, 304, 306, 570	\SOuL@octets	
\SOuL@addthreooctets	334, 432	. 325, 328, 374, 401, 405, 409, 411	
\SOUL@addtoken		\SOuL@orgDeclareOption	96, 101
. 426, 427, 448, 449, 456, 457, 595		\SOuL@orgDeclareRobustCommand	94, 99
\SOuL@addtwooctets	332, 416	\SOuL@orgnewcommand	95, 100
\SOuL@analyzeutfviii	324, 361	\SOuL@orgPackageError	97, 102
\SOuL@analyzeutfviiix	326, 385	\SOuL@orgSOUL@error	108, 112
\SOUL@apo	168,	\SOUL@postamble	722, 763
182, 253, 267, 487, 493, 495, 501		\SOUL@preamble	720, 732, 756
\SOuL@AtEnd	60, 61, 614	\SOuL@redefine	138, 158, 459,
\SOuL@charhash	357, 390, 395	469, 486, 503, 520, 545, 570, 595	
\SOUL@charkern	750	\SOuL@remainingtoken	464, 466
\SOuL@charthree	359, 404	\SOuL@restorelatexcmds	98, 105
\SOuL@chartwo	358, 408	\SOUL@scan	
\SOuL@checkutfviii	364, 369	. 159, 196, 204, 226, 244, 281,	
\SOuL@checkutfviiix	388, 394	289, 311, 423, 439, 445, 531,	
\SOuL@cmd	140, 145, 147, 156	543, 556, 568, 581, 593, 603, 612	
\SOuL@cmdcheck	141, 144	\SOUL@sethyphenchar	741
\SOUL@comma	166,	\SOUL@setkern	740, 747, 750
180, 251, 265, 470, 476, 478, 484		\SOUL@setup	731
\SOuL@defsanitizedstring	342, 346,	\SOUL@spaceskip	225, 310
347, 348, 349, 350, 351, 352, 353		\SOUL@spc	160, 197, 245, 282
\SOUL@dotoken	239, 329	\SOUL@splittoken	459
\SOUL@doword		\SOUL@stop	190, 275, 459, 466
. 192, 198, 206, 211, 215, 224,		\SOuL@stringcsnameu	391, 396
229, 232, 277, 283, 291, 296,		\SOuL@stringemdash	355, 441
300, 309, 314, 317, 420, 436, 442		\SOuL@stringendash	354, 435
\SOuL@empty	360, 400	\SOuL@stringfour	379
\SOUL@error	107	\SOuL@stringmacrocolon	390, 395
\SOUL@eval	158	\SOuL@stringnobreakspace	356, 419
\SOUL@eventuallyxhyphen		\SOuL@stringoctets	366, 370
. 193, 199, 207, 212, 216, 278,		\SOuL@stringthree	377
284, 292, 297, 301, 421, 437, 443		\SOuL@stringtwo	375
\SOUL@everyxhyphen	724, 746, 769	\SOuL@stringundeferr	391, 396
\SOUL@everyhyphen	723, 738, 766	\SOuL@stringUTFviii	366, 370
\SOUL@everylowerthan	727, 778	\SOUL@syllable	737
\SOUL@everyspace		\SOuL@temp	142, 145, 147,
. 202, 208, 213, 233, 287,		149, 152, 373, 375, 377, 379,	
293, 298, 318, 422, 721, 736, 760		399, 400, 403, 404, 407, 408,	
\SOUL@everysyllable	725, 737, 772	417, 418, 419, 433, 434, 435, 441	
\SOUL@everytoken	726, 775	\SOUL@token	460, 463, 726
\SOUL@exhyphen	217, 302, 438, 444	\SOUL@word	461, 467, 472, 474, 480,
\SOUL@flushapo		482, 489, 491, 497, 499, 506,	
. 169, 175, 188, 254, 260, 273, 486		508, 514, 516, 524, 525, 536,	
\SOUL@flushcomma		537, 549, 550, 561, 562, 574,	
. 167, 174, 187, 252, 259, 272, 469		575, 586, 587, 597, 598, 606, 607	
\SOUL@flushgrave		\SOULH	757
. 171, 176, 189, 256, 261, 274, 503		\soulH	755
\SOUL@flushminus		\soulHadd	758, 761,
. 165, 173, 186, 250, 258, 271		764, 767, 770, 773, 776, 779, 783	

<code>\soulHstring</code>	785,	<code>\TMP@EnsureCode</code>	59,
	786, 804, 806, 810, 813, 817, 827		66, 67, 68, 69, 70, 71, 72, 73,
<code>\soulomit</code>	222, 223, 307, 308, 935, 998		74, 75, 76, 77, 78, 79, 80, 81,
<code>\space</code>	736, 880, 908, 919		82, 83, 84, 85, 86, 87, 88, 89, 90, 91
<code>\spaceskip</code>	225, 310	<code>\tracingonline</code>	836
<code>\ss</code>	979, 982	<code>\tw@</code> 166, 168, 170, 251, 253, 255, 376, 411	
<code>\strip@prefix</code>	119, 122		
<code>\subsection</code>	797		
<code>\sy</code>	730		
	T		U
<code>\Test</code>	637, 655, 929, 974, 975, 976,	<code>\UNDEFINED</code>	660, 1005
	981, 982, 983, 984, 985, 986,	<code>\unexpanded</code>	660
	987, 988, 989, 990, 991, 992,	<code>\unskip</code>	208, 293
	993, 994, 995, 996, 997, 998,	<code>\usepackage</code> 663, 664, 665, 666, 668,	
	999, 1000, 1001, 1002, 1003, 1012	669, 670, 671, 706, 708, 713, 790	
<code>\TestDriver</code>	796, 963,		V
	964, 965, 966, 967, 968, 969, 970	<code>\v</code>	986
<code>\testdriver</code>	799, 805, 812, 826, 864	<code>\vbox</code>	861
<code>\TestInfo</code>	830, 831, 832, 867	<code>\vrule</code>	908, 919
<code>\TestInfoVBox</code>	833, 905		W
<code>\teststringA</code>	800, 805, 807	<code>\wd</code>	839, 840,
<code>\teststringB</code>	801, 812, 814, 826, 828		847, 848, 874, 875, 880, 886, 895
<code>\textbackslash</code>		<code>\write</code>	20, 41
	933, 934, 936, 939, 941, 943, 946		
<code>\textbraceleft</code>	936, 939, 943, 946		X
<code>\textbraceright</code>	936, 939, 943, 946	<code>\x</code>	8, 10, 12, 19,
<code>\texteuro</code>	981		23, 25, 40, 45, 52, 362, 372, 386,
<code>\texttt</code>	721, 942, 999		398, 472, 474, 480, 482, 489,
<code>\the</code>	56, 62, 472, 474, 480, 482,		491, 497, 499, 506, 508, 514,
	489, 491, 497, 499, 506, 508,		516, 523, 529, 535, 542, 548,
	514, 516, 525, 537, 550, 562,		554, 560, 567, 573, 579, 585,
	575, 587, 598, 607, 623, 726,		592, 596, 602, 605, 611, 959, 962
	737, 839, 845, 846, 847, 848,		
	849, 855, 856, 857, 880, 886, 895		Z
<code>\thr@@</code>	164, 249, 378, 409	<code>\z@</code>	382, 401, 476,
			484, 493, 501, 510, 518, 733, 734