

# 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

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

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

## 1 Documentation

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

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

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

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

### 1.1 Patch

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

- A problem with additional levels of curly braces is fixed. As advantage more implicate kernings are detected. However, the result may be incompatible with the original behaviour of package `soul` because of these respected implicate kernings.
- $\varepsilon$ -T<sub>E</sub>X, especially `\unexpanded` is supported. This allows a better protection of token groups (`\mbox{...}`, `math, ...`).

### 1.2 Future

Currently package `soul` does not seem to be maintained. Nevertheless if there will be a new version that adds support for UTF-8, then this package may become obsolete.

## 2 Implementation

```
1 <*package>
```

### 2.1 Reload check and package identification

Reload check, especially if the package is not used with L<sup>A</sup>T<sub>E</sub>X.

```
2 \begingroup
3 \catcode44 12 % ,
4 \catcode45 12 % -
5 \catcode46 12 % .
6 \catcode58 12 % :
7 \catcode64 11 % @
8 \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\relax
54             \xdef#1{#3}%
55         \fi
56     }%
57 \fi
58 \expandafter\x\csname ver@soulutf8.sty\endcsname
59 \ProvidesPackage{soulutf8}%
60 [2007/09/09 v1.0 Adding support for UTF-8 to soul (H0)]

```

## 2.2 Catcodes

```

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

```

```

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

```

## 2.3 Loading packages

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

```
110 \ifx\documentclass\@undefined
```

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

First we check, whether package soul is already loaded.

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

In case of plain-T<sub>E</sub>X package soul defines some macros in a simple manner that will break the definitions of minilts.tex, for example. Therefore these macros are first saved and restored afterwards.

```

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

```

```

121    }%
122    \input soul.sty\relax
123    \SOuL@restorelatexcmds
124    \fi

```

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

```

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

133    \input etexcmds.sty\relax

```

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

```

134    \expandafter\ifx\csname @onelevel@sanitize\endcsname\relax
135        \def\@onelevel@sanitize#1{%
136            \edef#1{%
137                \expandafter\strip@prefix\meaning#1%
138            }%
139        }%

```

`\strip@prefix`

```

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

141    \fi
142    \else

```

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

```

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

```

### 2.3.3 $\epsilon$ -T<sub>E</sub>X

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

```

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

```

## 2.4 Macro for redefinitions

`\SOuL@redefine`

```

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

```

\SOUL@cmdcheck

```

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

```

## 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.

```

176 \SOUL@redefine\SOUL@eval{%

```

First the expected definition.

```

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

```



```

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

```



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

## 2.6 UTF-8 analysis

```

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

```

\S0uL@analyzeutfviii

9

```

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

```

### 2.6.3 Support for utf8x.def

\Soul@analyzeutfviiiix

```

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

```

\Soul@checkutfviiiix

```

412 \def\noexpand\Soul@checkutfviiiix
413 ##1\Soul@stringmacrocolon\Soul@charhash1##2##3##4##5##6%
414 \Soul@stringcsnameu##7\Soul@stringundeferr##8\noexpand\@nil
415 }%
416 \x{%
417 \def\Soul@temp{#7}%
418 \ifx\Soul@temp\Soul@empty
419 \chardef\Soul@octets=\z@
420 \else
421 \def\Soul@temp{#5}%
422 \ifx\Soul@temp\Soul@charthree
423 \chardef\Soul@octets=4 %
424 \else
425 \def\Soul@temp{#3}%
426 \ifx\Soul@temp\Soul@chartwo
427 \chardef\Soul@octets=\thr@@
428 \else
429 \chardef\Soul@octets=\tw@
430 \fi
431 \fi
432 \fi
433 }

```

## 2.7 Actions for UTF-8 sequences

\Soul@addtwooctets

```

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

```

```

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

```

**\Soul@addthreeoctets**

```

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

```

**\Soul@addfouroctets**

```

473 \def\Soul@addfouroctets#1#2#3#4{%
474 ! \Soul@addtoken{\noexpand#1\noexpand#2\noexpand#3\noexpand#4}}%
475 + \Soul@addtoken{\etex@unexpanded{#1#2#3#4}}}%
476 }

```

## 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.

```

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

```

**\Soul@remainingtoken**

```

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

```

## 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`

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

`\SOUL@flushapo`

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

`\SOUL@flushgrave`

```
521 \SOUL@redefine\SOUL@flushgrave{%
522   \ifcase\SOUL@grave
523   \or
524     \edef\x{\SOUL@word={\the\SOUL@word'}}\x
525   \or
526     \edef\x{\SOUL@word={\the\SOUL@word{' '}}}\x
527   \fi
528   \SOUL@grave\z@
529 }{%
530   \ifcase\SOUL@grave
531   \or
532     \edef\x{\SOUL@word={\the\SOUL@word'}}\x
533   \or
534     \edef\x{\SOUL@word={\the\SOUL@word{' '}}}\x
535   \fi
536   \SOUL@grave\z@
537 }
```

\SOUL@addgroup

```

538 \SOuL@redefine\SOUL@addgroup#1#2{%
539   {%
540     \let\protect\noexpand
541     \edef\x{%
542       \global\SOUL@word={%
543         \the\SOUL@word
544         {{\noexpand#1#2}}}%
545     }%
546   }%
547   \x
548 }%
549 \SOUL@scan
550 }#1#2{%
551   \begingroup
552   \let\protect\noexpand
553   \edef\x{\endgroup
554     \SOUL@word={%
555       \the\SOUL@word
556       !      {\noexpand#1{#2}}}%
557   +      {\etex@unexpanded{#1{#2}}}}%
558   }%
559 }%
560 \x
561 \SOUL@scan
562 }

```

\SOUL@addmath

```

563 \SOuL@redefine\SOUL@addmath$#1${%
564   {%
565     \let\protect\noexpand
566     \edef\x{%
567       \global\SOUL@word={%
568         \the\SOUL@word
569         {{\hbox{ $#1$ }}}}%
570     }%
571   }%
572   \x
573 }%
574 \SOUL@scan
575 }$#1${%
576   \begingroup
577   \let\protect\noexpand
578   \edef\x{\endgroup
579     \SOUL@word={%
580       \the\SOUL@word
581       !      {\hbox{ $#1$ }}}}%
582   +      {\etex@unexpanded{\hbox{ $#1$ }}}}%
583   }%
584 }%
585 \x
586 \SOUL@scan
587 }

```

\SOUL@addprotect

```

588 \SOuL@redefine\SOUL@addprotect#1#2{%
589   {%
590     \let\protect\noexpand
591     \edef\x{%
592       \global\SOUL@word={%
593         \the\SOUL@word
594         {{\hbox{ #2 }}}}%

```

```

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

```

\SOUL@addtoken

```

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

632 \SOUL@AtEnd
633 </package>

```

## 3 Test

### 3.1 Catcode checks for loading

```

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

```

```

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

```

```

710 \end
711 </test1>

```

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

```

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

```

Input encoding utf8.

```

760 <*utf8>
761 \usepackage[utf8]{inputenc}
762 \DeclareUnicodeCharacter{00A0}{\nobreakspace}
763 </utf8>

```

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

```

764 <*utf8x>
765 \let\origshow\show
766 \usepackage[utf8x]{inputenc}

```



```

767 \AtBeginDocument{%
768   \let\show\origshow
769 }
770 </utf8x>

```

### 3.2.1 Driver definitions

```

771 \makeatletter

```

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

```

772 \DeclareRobustCommand*\an{%
773   \def\Soul@preamble{$\sim$P}%
774   \def\Soul@everyspace##1{##1\texttt{\char'\linebreak[2]}}%
775   \def\Soul@postamble{$\sim$E}%
776   \def\Soul@everyhyphen{$\sim$}\linebreak[1]}%
777   \def\Soul@everyexhyphen##1{##1$\sim$}%
778   \def\Soul@everysyllable{$\sim$S}%
779   \def\Soul@everytoken{\the\Soul@token$\sim$T}%
780   \def\Soul@everylowerthan{$\sim$L}%
781   \Soul@
782 }

```

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

```

783 \DeclareRobustCommand*\sy{%
784   \Soul@setup
785   \def\Soul@preamble{%
786     \lefthyphenmin=\z@
787     \righthyphenmin=\z@
788   }%
789   \def\Soul@everyspace##1{##1\space}%
790   \def\Soul@everysyllable{\the\Soul@syllable}%
791   \def\Soul@everyhyphen{%
792     \discretionary{%
793       \Soul@setkern\Soul@hyphkern
794       \Soul@sethyphenchar
795     }{}%
796     \hbox{\kern1pt$\cdot$}%
797   }%
798   \def\Soul@everyexhyphen##1{%
799     \Soul@setkern\Soul@hyphkern
800     \hbox{##1}%
801     \discretionary{}{}%
802     \Soul@setkern\Soul@charkern
803   }%
804   \def\Soul@
805 }

```

\soulH

```

808 \DeclareRobustCommand*\soulH{%
809   \def\Soul@preamble{%
810     \gdef\SoulH{}%
811     \soulHadd{}%
812   }%
813   \def\Soul@everyspace##1{%
814     \soulHadd{Sp}%
815   }%
816   \def\Soul@postamble{%
817     \soulHadd{}}%
818   \def\Soul@everyhyphen{%
819     \soulHadd{Hy}%

```

```

821 }%
822 \def\SOUL@everyexhyphen##1{%
823   \soulHadd{Exhy}%
824 }%
825 \def\SOUL@everysyllable{%
826   \soulHadd{Sy}%
827 }%
828 \def\SOUL@everytoken{%
829   \soulHadd{T}%
830 }%
831 \def\SOUL@everylowerthan{%
832   \soulHadd{<}%
833 }%
834 \SOUL@
835 }
836 \newcommand*{\soulHadd}[1]{%
837   #1%
838   \expandafter\gdef\expandafter\soulHstring\expandafter{%
839     \soulHstring
840     #1%
841   }%
842 }

```

### 3.2.2 Test macros

```

843 \usepackage{qstest}
844 \IncludeTests{*}
845 \LogTests{log}{*}{*}

846 \newsavebox\boxA
847 \newsavebox\boxB
848 \newsavebox\boxC

\TestDriver

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

```

```

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

```

\SetTestVBox

```

913 \newcommand*{\SetTestVBox}[2]{%
914 \setbox#1=\vbox{%
915 \hsize=0pt\relax
916 \hfuzz=100mm\relax
917 \expandafter\testdriver\expandafter{#2}%
918 }%
919 }

```

\TestInfo

```

920 \newcommand*{\TestInfo}[2]{%
921 \par
922 \noindent
923 \copy#1\relax
924 \hfill
925 \begin{group}
926 \small
927 \ifdim\wd0=\wd2\relax
928 \ifdim\wd0=\wd4\relax
929 \else
930 \ifnum#1=2 %
931 \raisebox{.5\baselineskip}[0pt][0pt]{%
932 \llap{$\}$}%
933 \rlap{width = \the\wd#1\relax\space$\}$}%
934 \phantom{width = 999.99999pt}%
935 \qqquad

```

```

936         }%
937     \else
938         \ifnum#1=4 %
939             \rlap{width = \the\wd#1}%
940             \phantom{width = 999.99999pt}%
941             \qqquad
942         \fi
943     \fi
944 \fi
945 \else
946     \begingroup
947     \bfseries
948     \rlap{width = \the\wd#1}%
949     \phantom{width = 999.99999pt}%
950 \endgroup
951 \qqquad
952 \fi
953 \rlap{(#2)}%
954 \phantom{(original soul)}%
955 \endgroup
956 \par
957 }

```

\TestInfoVBox

```

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

```

\Test

```

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

```

```

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

```

### 3.2.3 Testing

```

1026 \begin{document}
1027 \Test A{}{}%
1028 \Test B{Hello World}{Hello World}%
1029 \Test C{%
1030 Umlauts: ^^c3^^84^^c3^^96^^c3^^9c^^c3^^a4^^c3^^b6^^c3^^bc^^c3^^9f%
1031 }{%
1032 Umlauts: \"A\"O\"U\"a\"o\"u\"ss
1033 }%
1034 \Test D{Euro: ^^e2^^82^^ac}{Euro: \texteuro}%
1035 \Test E{Stra^^c3^^9fe}{Stra\ss e}%
1036 \Test F{R^^c3^^b6mer}{R\"omer}%
1037 \Test G{A^^e2^^80^^93B^^e2^^80^^94C}{A--B---C}%
1038 \Test H{X^^c2^^a0Y}{X~Y}%
1039 \Test I{A^^c5^^a4}{A\v T}%
1040 \Test J{A,B,,C,,,D,,,E}{A,B,,C,,,D,,,E}%
1041 \Test K{A'B'C''D'''E}{A'B'C''D'''E}%
1042 \Test L{A'B'C'C'D''E}{A'B'C'C'D''E}%
1043 \Test M{A$x$B}{A$x$B}%
1044 \Test N{A\mbox{abc}B}{A\mbox{abc}B}%
1045 \Test O{, ,TA''}{, ,TA''}%
1046 \Test P{{!'}Hola{?'}}{{!'}Hola{?'}}%
1047 \Test Q{re-sent\slash x}{re-sent\slash x}%
1048 \Test R{new\\line}{new\\line}%
1049 \Test S{foo\linebreak bar}{foo\linebreak bar}%
1050 \Test T{Virchow{sche}}{Virchow{sche}}%
1051 \Test U{A\soulomit{foo}B}{A\soulomit{foo}B}%
1052 \Test V{A\texttt{xyz}B}{A\texttt{xyz}B}%
1053 \Test W{ffi}{ffi}%

```

```

1054 \Test X{f{}f{}i}{f{}f{}i}%
1055 \Test Y{f\mbox{}f\mbox{}i}{f\mbox{}f\mbox{}i}%
1056 \Test Z{c{"a"}{c{"a"}}}%
1057 <*test2 | test3>
1058 \let\abcx\UNDEFINED
1059 \def\abc{%
1060     \begingroup
1061         \def\abcx{x}%
1062         \abcx
1063     \endgroup
1064 }%
1065 \Test{AA}{[{x\abc x}][{xxx}]}%
1066 </test2 | test3>
1067 \end{document}
1068 </test2 | test3 | test4 | test5>

```

## 4 Installation

### 4.1 Download

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

[CTAN:macros/latex/contrib/oberdiek/soulutf8.dtx](#) The source file.

[CTAN:macros/latex/contrib/oberdiek/soulutf8.pdf](#) Documentation.

**Bundle.** All the packages of the bundle ‘oberdiek’ are also available in a TDS compliant ZIP archive. There the packages are already unpacked and the documentation files are generated. The files and directories obey the TDS standard.

[CTAN:macros/latex/contrib/oberdiek/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/

```

### 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
```

---

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

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

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

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

## 4.4 Refresh file name databases

If your  $\text{\TeX}$  distribution (te $\text{\TeX}$ , mi $\text{\TeX}$ , ...) relies on file name databases, you must refresh these. For example, te $\text{\TeX}$  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  $\text{\LaTeX}$ .** The `.dtx` chooses its action depending on the format:

**plain- $\text{\TeX}$ :** Run `docstrip` and extract the files.

**$\text{\LaTeX}$ :** Generate the documentation.

If you insist on using  $\text{\LaTeX}$  for `docstrip` (really, `docstrip` does not need  $\text{\LaTeX}$ ), 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 $\text{\LaTeX}$ :

```
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	
\ " .....	1032, 1036, 1056
\ # .....	637, 699
\ % .....	702
\ @ .....	638, 695
\@PackageError .....	129
\@PackageInfo .....	851, 856, 862, 869
\@PackageWarningNoLine .....	166
\@ehc .....	730
\@empty .....	857, 863, 870
\@firstofone .....	646, 649
\@gobble .....	485, 643, 651
\@ifDefined .....	726, 735, 742, 743
\@ifdefinable .....	734
\@ifundefined .....	727
\@latex@error .....	728
\@ne .....	196, 198, 200,
	202, 281, 283, 285, 287, 889, 1009
\@nil .....	385, 388, 410, 414
\@onelevel@sanitize .....	<u>134</u> , 362, 436, 452
\@undefined .....	110
\[ .....	700
\ \ .....	223, 308, 696, 987, 1048
\{ .....	635, 697, 933
\} .....	636, 698, 932
\] .....	701
\~ .....	1005
\_ .....	703, 774
A	
\abc .....	994, 1059, 1065
\abcx .....	1058, 1061, 1062
\active .....	103
\advance .....	196, 198, 200, 202,
	281, 283, 285, 287, 676, 684, 1009
B	
\afterassignment .....	159
\an .....	<u>772</u>
\AtBeginDocument .....	767
C	
\baselineskip .....	931
\begin .....	855, 1026
\bfseries .....	947
\body .....	655, 659
\boxA .....	846, 860, 900, 901, 902,
	903, 905, 908, 909, 910, 961, 962
\boxB .....	847, 867,
	900, 902, 903, 906, 909, 962, 972
\boxC .....	848, 881, 901, 908, 910
\break ....	226, 246, 248, 311, 331, 333
C	
\c .....	998, 1056
\catcode .....	3, 4, 5, 6, 7, 18,
	19, 20, 34, 35, 36, 37, 38, 39, 40,
	41, 42, 43, 44, 62, 63, 66, 67, 68,
	69, 73, 74, 75, 76, 80, 82, 150,
	151, 153, 154, 635, 636, 637,
	638, 673, 682, 695, 696, 697,
	698, 699, 700, 701, 702, 703, 704
\cdot .....	796
\char .....	774
\chardef .....	392, 419, 423, 427, 429
\clearpage .....	984
\copy .....	923, 961, 972
\count@ .....	640,
	669, 673, 675, 676, 680, 682,
	683, 684, 1002, 1004, 1005, 1009
\countdef .....	640
\csname .....	8, 21, 45,
	58, 65, 111, 134, 361, 362, 639,



642, 645, 648, 687, 709, 720, 729, 736, 737, 744, 745, 852, 1012	\input ..... 122, 125, 133, 688
\CurrentOption ..... 143	\iterate ..... 656, 658, 660
<b>D</b>	<b>K</b>
\DeclareOption ..... 114, 119, 143	\kern ..... 796
\DeclareRobustCommand ..... ..... 112, 117, 772, 783, 808	<b>L</b>
\DeclareUnicodeCharacter ..... 762	\lccode ..... 1005
\discretionary ..... 792, 802	\leavevmode ..... 214, 299
\documentclass ..... 110, 715	\lefthyphenmin ..... 786
\dp ..... 899, 909, 910	\linebreak ..... 249, 251, 334, 336, 774, 776, 986, 1049
<b>E</b>	\llap ..... 932
\empty ..... 12	\LoadCommand ..... 688, 705
\end ..... 710, 911, 1067	\LogTests ..... 845
\endcsname ..... 8, 21, 45, 58, 65, 111, 134, 361, 362, 639, 642, 645, 648, 687, 709, 720, 729, 736, 737, 744, 745, 852, 1012	\loop ..... 654, 670, 681, 1003
\endinput ..... 30	\lowercase ..... 1006
\etex@unexpanded ..... . 445, 467, 475, 557, 582, 607, 626	<b>M</b>
\Expect ..... 890, 891, 892, 898, 899, 900, 901, 902, 908, 909, 910	\makeatletter ..... 725, 771, 982
<b>G</b>	\makeatother ..... 749, 1025
\gdef ..... 810, 838	\maxdimen ..... 887, 888
<b>H</b>	\mbox 236, 321, 961, 972, 991, 1044, 1055
\hbox .. 238, 323, 569, 581, 582, 594, 606, 607, 796, 801, 858, 865, 879	\meaning ..... 137, 383, 407
\hfil ..... 226, 311	\MessageBreak ..... 167, 168, 169
\hfill ..... 924, 964, 973	<b>N</b>
\hfuzz ..... 916	\NeedsTeXFormat ..... 714
\hsize ..... 915	\newcommand .... 113, 118, 726, 733, 741, 836, 849, 913, 920, 958, 983
\HstringA ..... 859, 890, 891	\newsavebox ..... 846, 847, 848
\HstringB ..... 866, 890	\next ..... 660, 662, 664
\HstringC ..... 880, 891	\nobreak ..... 226, 231, 311, 316, 440
\ht ..... 898, 902, 903, 908, 962	\nobreakspace ..... 762
<b>I</b>	\noindent ..... 922, 960, 971
\if ..... 178, 215, 263, 300	\null ..... 211, 217, 225, 230, 296, 302, 310, 315, 439
\ifcase ..... 9, 343, 346, 488, 496, 505, 513, 522, 530	<b>O</b>
\ifcat ..... 252, 254, 337, 339	\origshow ..... 765, 768
\ifdim ..... 893, 903, 927, 928, 962	<b>P</b>
\ifetex@unexpanded ..... 149	\PackageError ..... 115, 120, 129
\ifnum . 182, 184, 186, 188, 267, 269, 271, 273, 675, 683, 930, 938, 1004	\PackageInfo ..... 26
\ifSOUL@ignorespaces ..... 218, 303	\par ..... 213, 214, 298, 299, 921, 956, 959, 970, 980
\ifx ..... 10, 12, 21, 45, 53, 110, 111, 134, 163, 191, 192, 193, 194, 195, 197, 199, 201, 208, 213, 223, 228, 232, 236, 238, 240, 246, 249, 276, 277, 278, 279, 280, 282, 284, 286, 293, 298, 308, 313, 317, 321, 323, 325, 331, 334, 393, 395, 397, 418, 422, 426, 437, 453, 459, 639, 642, 645, 648, 687	\PassOptionsToPackage ..... 143
\immediate ..... 23, 47	\phantom .. 934, 940, 949, 954, 968, 977
\IncludeTests ..... 844	\ProcessOptions ..... 144
	\protect .. 540, 552, 565, 577, 590, 602
	\ProvidesPackage ..... 59
	<b>Q</b>
	\qqquad ..... 935, 941, 951
	<b>R</b>
	\raisebox ..... 931
	\RangeCatcodeInvalid ..... ..... 679, 691, 692, 693, 694
	\repeat ..... 654, 666, 677, 685, 1010
	\RequirePackage ..... 145, 146, 147
	\RestoreCatcodes .. 668, 671, 672, 706
	\RestoreOrg ..... 741, 864, 871, 872, 873, 874, 875, 876, 877, 878

\righthyphenmin	787	\SOUL@everyospace	220, 226, 231, 251, 305, 311, 316, 336, 440, 774, 789, 813
\rlap	933, 939, 948, 953, 967, 976	\SOUL@everyosyllable	778, 790, 825
<b>S</b>			
\SaveOrg	733, 750, 751, 752, 753, 754, 755, 756, 757, 758	\SOUL@everytoken	779, 828
\section	1013	\SOUL@exhyphen	235, 320, 456, 462
\setbox	858, 865, 879, 914	\SOUL@flushapo	187, 193, 206, 272, 278, 291, 504
\SetTestVBox	860, 867, 881, 913	\SOUL@flushcomma	185, 192, 205, 270, 277, 290, 487
\show	765, 768	\SOUL@flushgrave	189, 194, 207, 274, 279, 292, 521
\showbox	895, 896, 905, 906	\SOUL@flushminus	183, 191, 204, 268, 276, 289
\showboxbreadth	888	\SOUL@grave	188, 202, 273, 287, 522, 528, 530, 536
\showboxdepth	887	\SOUL@hyphkern	793, 800
\slash	232, 317, 1047	\SOUL@ignorespacesfalse	180, 265
\small	926, 966, 975	\SOUL@ignorespacestrue	227, 312
\SOUL@	781, 806, 834	\SOUL@minus	182, 196, 267, 281
\SOUL@@	178, 191, 192, 193, 194, 195, 197, 199, 201, 208, 213, 215, 223, 228, 232, 236, 238, 240, 246, 249, 252, 254, 263, 276, 277, 278, 279, 280, 282, 284, 286, 293, 298, 300, 308, 313, 317, 321, 323, 325, 331, 334, 337, 339, 383, 407	\SOUL@on	177, 209, 214, 222, 237, 239, 241, 253, 255, 257, 260, 262, 294, 299, 307, 322, 324, 326, 338, 340, 347, 350, 352, 354, 358
\SOuL@addfouroctets	354, 473	\SOuL@next	441, 443, 448, 457, 463, 465, 471
\SOUL@addgroup	253, 338, 538	\SOuL@octets	343, 346, 392, 419, 423, 427, 429
\SOUL@addmath	255, 340, 563	\SOuL@orgDeclareOption	114, 119
\SOUL@addprotect	237, 239, 322, 324, 588	\SOuL@orgDeclareRobustCommand	112, 117
\SOuL@addthreooctets	352, 450	\SOuL@orgnewcommand	113, 118
\SOUL@addtoken	444, 445, 466, 467, 474, 475, 613	\SOuL@orgPackageError	115, 120
\SOuL@addtwooctets	350, 434	\SOuL@orgSOUL@error	126, 130
\SOuL@analyzeutfviii	342, 379	\SOUL@postamble	775, 816
\SOuL@analyzeutfviiix	344, 403	\SOUL@preamble	773, 785, 809
\SOUL@apo	186, 200, 271, 285, 505, 511, 513, 519	\SOuL@redefine	156, 176, 477, 487, 504, 521, 538, 563, 588, 613
\SOuL@AtEnd	78, 79, 632	\SOuL@remainingtoken	482, 484
\SOuL@charhash	375, 408, 413	\SOuL@restorelatexcmds	116, 123
\SOUL@charkern	803	\SOUL@scan	177, 214, 222, 244, 262, 299, 307, 329, 441, 457, 463, 549, 561, 574, 586, 599, 611, 621, 630
\SOuL@charthree	377, 422	\SOUL@sethyphenchar	794
\SOuL@chartwo	376, 426	\SOUL@setkern	793, 800, 803
\SOuL@checkutfviii	382, 387	\SOUL@setup	784
\SOuL@checkutfviiix	406, 412	\SOUL@spaceskip	243, 328
\SOuL@cmd	158, 163, 165, 174	\SOUL@spc	178, 215, 263, 300
\SOuL@cmdcheck	159, 162	\SOUL@splittoken	477
\SOUL@comma	184, 198, 269, 283, 488, 494, 496, 502	\SOUL@stop	208, 293, 477, 484
\SOuL@defsanitizedstring	360, 364, 365, 366, 367, 368, 369, 370, 371	\SOuL@stringcsnameu	409, 414
\SOUL@dotoken	257, 347	\SOuL@stringemdash	373, 459
\SOUL@doword	210, 216, 224, 229, 233, 242, 247, 250, 295, 301, 309, 314, 318, 327, 332, 335, 438, 454, 460	\SOuL@stringendash	372, 453
\SOuL@empty	378, 418	\SOuL@stringfour	397
\SOUL@error	125	\SOuL@stringmacrocolon	408, 413
\SOUL@eval	176	\SOuL@stringnobreakspace	374, 437
\SOUL@eventuallyexhyphen	211, 217, 225, 230, 234, 296, 302, 310, 315, 319, 439, 455, 461	\SOuL@stringoctets	384, 388
\SOUL@everyexhyphen	777, 799, 822	\SOuL@stringthree	395
\SOUL@everyhyphen	776, 791, 819	\SOuL@stringtwo	393
\SOUL@everylowerthan	780, 831	\SOuL@stringundeferr	409, 414
		\SOuL@stringUTFviii	384, 388

<code>\SOUL@syllable</code> . . . . .	790	<code>\texttt</code> . . . . .	774, 995, 1052
<code>\SOuL@temp</code> . . . . .	160, 163, 165, 167, 170, 391, 393, 395, 397, 417, 418, 421, 422, 425, 426, 435, 436, 437, 451, 452, 453, 459	<code>\the</code> . . . . .	66, 67, 68, 69, 80, 490, 492, 498, 500, 507, 509, 515, 517, 524, 526, 532, 534, 543, 555, 568, 580, 593, 605, 616, 625, 673, 779, 790, 892, 898, 899, 900, 901, 902, 908, 909, 910, 933, 939, 948
<code>\SOUL@token</code> . . . . .	478, 481, 779	<code>\thr@@</code> . . . . .	182, 267, 396, 427
<code>\SOUL@word</code> . . . . .	479, 485, 490, 492, 498, 500, 507, 509, 515, 517, 524, 526, 532, 534, 542, 543, 554, 555, 567, 568, 579, 580, 592, 593, 604, 605, 615, 616, 624, 625	<code>\TMP@EnsureCode</code> . . . . .	77, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109
<code>\SOULH</code> . . . . .	810	<code>\tracingonline</code> . . . . .	889
<code>\soulH</code> . . . . .	<u>808</u>	<code>\tw@</code> . . . . .	184, 186, 188, 269, 271, 273, 394, 429
<code>\soulHadd</code> . . . . .	811, 814, 817, 820, 823, 826, 829, 832, 836		
<code>\soulHstring</code> . . . . .	838, 839, 857, 859, 863, 866, 870, 880		
<code>\soulomit</code> . . . . .	240, 241, 325, 326, 988, 1051		
<code>\space</code> . . . . .	789, 933, 961, 972		
<code>\spaceskip</code> . . . . .	243, 328		
<code>\ss</code> . . . . .	1032, 1035		
<code>\strip@prefix</code> . . . . .	137, <u>140</u>		
<code>\subsection</code> . . . . .	850		
<code>\sy</code> . . . . .	<u>783</u>		
	<b>T</b>		
<code>\Test</code> . . . . .	690, 708, <u>982</u> , 1027, 1028, 1029, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1065		
<code>\TestDriver</code> . . . . .	<u>849</u> , 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023		
<code>\testdriver</code> . . . . .	852, 858, 865, 879, 917		
<code>\TestInfo</code> . . . . .	883, 884, 885, <u>920</u>		
<code>\TestInfoVBox</code> . . . . .	886, <u>958</u>		
<code>\teststringA</code> . . . . .	853, 858, 860		
<code>\teststringB</code> . . . . .	854, 865, 867, 879, 881		
<code>\textbackslash</code> . . . . .	986, 987, 989, 992, 994, 996, 999		
<code>\textbraceleft</code> . . . . .	989, 992, 996, 999		
<code>\textbraceright</code> . . . . .	989, 992, 996, 999		
<code>\texteuro</code> . . . . .	1034		
		<b>U</b>	
		<code>\UNDEFINED</code> . . . . .	713, 1058
		<code>\unexpanded</code> . . . . .	713
		<code>\unskip</code> . . . . .	226, 311
		<code>\usepackage</code> . . . . .	716, 717, 718, 719, 721, 722, 723, 724, 759, 761, 766, 843
		<b>V</b>	
		<code>\v</code> . . . . .	1039
		<code>\vbox</code> . . . . .	914
		<code>\vrule</code> . . . . .	961, 972
		<b>W</b>	
		<code>\wd</code> . . . . .	892, 893, 900, 901, 927, 928, 933, 939, 948
		<code>\write</code> . . . . .	23, 47
		<b>X</b>	
		<code>\x</code> . . . . .	8, 10, 12, 22, 26, 28, 46, 51, 58, 64, 72, 380, 390, 404, 416, 490, 492, 498, 500, 507, 509, 515, 517, 524, 526, 532, 534, 541, 547, 553, 560, 566, 572, 578, 585, 591, 597, 603, 610, 614, 620, 623, 629, 1012, 1015
		<b>Z</b>	
		<code>\z@</code> . . . . .	400, 419, 494, 502, 511, 519, 528, 536, 786, 787