

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 | \LaTeX | 5 |
| 2.3.3 | ϵ - \TeX | 5 |
| 2.4 | Macro for redefinitions | 5 |
| 2.5 | Redefinition of <code>\SOUL@eval</code> | 6 |
| 2.6 | UTF-8 analysis | 9 |
| 2.6.1 | Help strings | 9 |
| 2.6.2 | Support for <code>utf8.def</code> | 9 |
| 2.6.3 | Support for <code>utf8x.def</code> | 10 |
| 2.7 | Actions for UTF-8 sequences | 10 |
| 2.7.1 | Redefinition of <code>\SOUL@splittoken</code> | 11 |
| 2.8 | Patches | 12 |
| 3 | Test | 14 |
| 3.1 | Catcode checks for loading | 14 |
| 3.2 | \LaTeX tests | 16 |
| 3.2.1 | Driver definitions | 17 |
| 3.2.2 | Test macros | 18 |
| 3.2.3 | Testing | 21 |
| 4 | Installation | 22 |
| 4.1 | Download | 22 |
| 4.2 | Bundle installation | 22 |
| 4.3 | Package installation | 22 |
| 4.4 | Refresh file name databases | 23 |
| 4.5 | Some details for the interested | 23 |
| 5 | References | 24 |

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

1 Documentation

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

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

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

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

1.1 Patch

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

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

1.2 Future

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

2 Implementation

```
1 <*package>
```

2.1 Reload check and package identification

Reload check, especially if the package is not used with L^AT_EX.

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

```

12     \ifx\x\empty % LaTeX
13     \else
14         1%
15     \fi
16 \fi
17 \else
18     \catcode35 6 % #
19     \catcode123 1 % {
20     \catcode125 2 % }
21     \expandafter\ifx\csname PackageInfo\endcsname\relax
22         \def\x#1#2{%
23             \immediate\write-1{Package #1 Info: #2.}%
24         }%
25     \else
26         \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
27     \fi
28     \x{soulutf8}{The package is already loaded}%
29     \endgroup
30     \expandafter\endinput
31 \fi
32 \endgroup

```

Package identification:

```

33 \begingroup
34 \catcode35 6 % #
35 \catcode40 12 % (
36 \catcode41 12 % )
37 \catcode44 12 % ,
38 \catcode45 12 % -
39 \catcode46 12 % .
40 \catcode47 12 % /
41 \catcode58 12 % :
42 \catcode64 11 % @
43 \catcode123 1 % {
44 \catcode125 2 % }
45 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
46     \def\x#1#2#3[#4]{\endgroup
47         \immediate\write-1{Package: #3 #4}%
48         \xdef#1{#4}%
49     }%
50 \else
51     \def\x#1#2[#3]{\endgroup
52         #2[#3]}%
53     \ifx#1\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^AT_EX.

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

2.3.1 plain-T_EX

First we check, whether package soul is already loaded.

```
111 \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.

```

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^AT_EX'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^AT_EX

```

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 ϵ -T_EX

In plain-T_EX 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}%
215   \fi
216 \fi
217 }
```



```

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@ignoreSPACEtrue
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@@

```



```

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@analyzeutfviii

```

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

```

\Soul@checkutfviii

```

412 \def\noexpand\Soul@checkutfviii
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 \def\Test{%
688 \RangeCatcodeInvalid{0}{47}%
689 \RangeCatcodeInvalid{58}{64}%
690 \RangeCatcodeInvalid{91}{96}%
691 \RangeCatcodeInvalid{123}{255}%
692 \catcode'\@=12 %
693 \catcode'\=0 %
694 \catcode'\{=1 %
695 \catcode'\}=2 %
696 \catcode'\#=6 %
697 \catcode'\[=12 %
698 \catcode'\]=12 %
699 \catcode'\%=14 %
700 \catcode'\ =10 %
701 \catcode13=5 %
702 \input soulutf8.sty\relax
703 \RestoreCatcodes
704 }
705 \Test
706 \csname @@end\endcsname
707 \end
708 </test1>

```

3.2 L^AT_EX tests

```

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

```

Input encoding utf8.

```

757 <*utf8>
758 \usepackage[utf8]{inputenc}
759 \DeclareUnicodeCharacter{00A0}{\nobreakspace}
760 </utf8>

```

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

```

761 <*utf8x>
762 \let\origshow\show
763 \usepackage[utf8x]{inputenc}
764 \AtBeginDocument{%
765   \let\show\origshow
766 }

```


767 `\utf8x`

3.2.1 Driver definitions

768 `\makeatletter`

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

```
769 \DeclareRobustCommand*{\an}{%
770   \def\SOUL@preamble{$^{~P}$}%
771   \def\SOUL@everySPACE##1{##1\texttt{\char'\ }\\linebreak[2]}}%
772   \def\SOUL@postamble{$^{~E}$}%
773   \def\SOUL@everyhyphen{$^{~}$\\linebreak[1]}}%
774   \def\SOUL@everyexhyphen##1{##1$^{~=}$}%
775   \def\SOUL@everysyllable{$^{~S}$}%
776   \def\SOUL@everytoken{\the\SOUL@token$^{~T}$}%
777   \def\SOUL@everylowerthan{$^{~L}$}%
778   \SOUL@
779 }
```

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

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

`\soulH`

```
805 \DeclareRobustCommand*{\soulH}{%
806   \def\SOUL@preamble{%
807     \gdef\SOULH{}%
808     \soulHadd{[]}%
809   }%
810   \def\SOUL@everySPACE##1{%
811     \soulHadd{Sp}%
812   }%
813   \def\SOUL@postamble{%
814     \soulHadd{[]}%
815   }%
816   \def\SOUL@everyhyphen{%
817     \soulHadd{Hy}%
818   }%
819   \def\SOUL@everyexhyphen##1{%
820     \soulHadd{Exhy}%
821   }%
822 }
```

```

821 }%
822 \def\SOUL@every syllable{%
823   \soulHadd{Sy}%
824 }%
825 \def\SOUL@every token{%
826   \soulHadd{T}%
827 }%
828 \def\SOUL@every lower than{%
829   \soulHadd{<}%
830 }%
831 \SOUL@
832 }
833 \newcommand*{\soulHadd}[1]{%
834   #1%
835   \expandafter\gdef\expandafter\soulHstring\expandafter{%
836     \soulHstring
837     #1%
838   }%
839 }

```

3.2.2 Test macros

```

840 \usepackage{qstest}
841 \IncludeTests{*}
842 \LogTests{log}{*}{*}

843 \newsavebox\boxA
844 \newsavebox\boxB
845 \newsavebox\boxC

\TestDriver

846 \newcommand*{\TestDriver}[4]{%
847   \subsection{Driver #1}%
848   \@PackageInfo{soulutf8-test}{* Driver #1}%
849   \expandafter\let\expandafter\testdriver\csname #1\endcsname
850   \def\teststringA{#3}%
851   \def\teststringB{#4}%
852   \begin{qstest}{string-#2-driver-#1}{string,#2,driver,#1}%
853     \@PackageInfo{soulutf8-test}{* Version: soulutf8}%
854     \global\let\soulHstring\@empty
855     \setbox0=\hbox{\expandafter\testdriver\expandafter{\teststringA}}%
856     \let\HstringA\soulHstring
857     \SetTestVBox{\boxA}{\teststringA}%
858     %
859     \@PackageInfo{soulutf8-test}{* Version: soul + patch}%
860     \global\let\soulHstring\@empty
861     \RestoreOrg{eval}%
862     \setbox2=\hbox{\expandafter\testdriver\expandafter{\teststringB}}%
863     \let\HstringB\soulHstring
864     \SetTestVBox{\boxB}{\teststringB}%
865     %
866     \@PackageInfo{soulutf8-test}{* Version: original soul}%
867     \global\let\soulHstring\@empty
868     \RestoreOrg{splittoken}%
869     \RestoreOrg{flushcomma}%
870     \RestoreOrg{flushapo}%
871     \RestoreOrg{grave}%
872     \RestoreOrg{addgroup}%
873     \RestoreOrg{addmath}%
874     \RestoreOrg{addprotect}%
875     \RestoreOrg{addtoken}%
876     \setbox4=\hbox{\expandafter\testdriver\expandafter{\teststringB}}%
877     \let\HstringC\soulHstring

```

```

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

```

\SetTestVBox

```

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

```

\TestInfo

```

917 \newcommand*{\TestInfo}[2]{%
918 \par
919 \noindent
920 \copy#1\relax
921 \hfill
922 \begingroup
923 \small
924 \ifdim\wd0=\wd2\relax
925 \ifdim\wd0=\wd4\relax
926 \else
927 \ifnum#1=2 %
928 \raisebox{.5\baselineskip}[Opt][Opt]{%
929 \llap{$\}$ }%
930 \rlap{width = \the\wd#1\relax\space$\{\}$}%
931 \phantom{width = 999.99999pt}}%
932 \qqquad
933 }%
934 \else
935 \ifnum#1=4 %

```

```

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

```

\TestInfoVBox

```

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

```

\Test

```

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

```

```

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

```

3.2.3 Testing

```

1023 \begin{document}
1024 \Test A{}{}%
1025 \Test B{Hello World}{Hello World}%
1026 \Test C{%
1027   Umlauts: ^^c3^^84^^c3^^96^^c3^^9c^^c3^^a4^^c3^^b6^^c3^^bc^^c3^^9f%
1028 }{%
1029   Umlauts: \ "A" "O" "U" "a" "o" "u" \ss
1030 }%
1031 \Test D{Euro: ^^e2^^82^^ac}{Euro: \texteuro}%
1032 \Test E{Stra^^c3^^9fe}{Stra\ss e}%
1033 \Test F{R^^c3^^b6mer}{R"omer}%
1034 \Test G{A^^e2^^80^^93B^^e2^^80^^94C}{A--B---C}%
1035 \Test H{X^^c2^^a0Y}{X~Y}%
1036 \Test I{A^^c5^^a4}{A\v T}%
1037 \Test J{A,B,,C,,,D,,,E}{A,B,,C,,,D,,,E}%
1038 \Test K{A'B''C'''D''''E}{A'B''C'''D''''E}%
1039 \Test L{A'B'C''D'''E}{A'B'C''D'''E}%
1040 \Test M{A$x$B}{A$x$B}%
1041 \Test N{A\mbox{abc}B}{A\mbox{abc}B}%
1042 \Test O{,TA''}{,TA''}%
1043 \Test P{{!'}Hola{?'}}{{!'}Hola{?'}}%
1044 \Test Q{re-sent\slash x}{re-sent\slash x}%
1045 \Test R{new\\line}{new\\line}%
1046 \Test S{foo\linebreak bar}{foo\linebreak bar}%
1047 \Test T{Virchow{sche}}{Virchow{sche}}%
1048 \Test U{A\soulomit{foo}B}{A\soulomit{foo}B}%
1049 \Test V{A\texttt{xyz}B}{A\texttt{xyz}B}%
1050 \Test W{ffi}{ffi}%
1051 \Test X{f{}f{}i}{f{}f{}i}%
1052 \Test Y{f\mbox{}f\mbox{}i}{f\mbox{}f\mbox{}i}%
1053 \Test Z{\c{"a}}{\c{"a}}%

```

```

1054 <*test2 | test3>
1055   \let\abcx\UNDEFINED
1056   \def\abc{%
1057     \begingroup
1058       \def\abcx{x}%
1059     \abcx
1060   \endgroup
1061 }%
1062   \Test{AA}{[{x\abc x}]}[{xxx}]}%
1063 </test2 | test3>
1064 \end{document}
1065 </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/
```

4.3 Package installation

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

```
tex soulutf8.dtx
```

¹<http://ftp.ctan.org/tex-archive/>

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

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

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

4.4 Refresh file name databases

If your \TeX distribution (te \TeX , mi \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 | | | |
|------------------------|---|------------------|---|
| \" | 1029, 1033, 1053 | \afterassignment | 159 |
| \# | 637, 696 | \an | <u>769</u> |
| \% | 699 | \AtBeginDocument | 764 |
| \@ | 638, 692 | B | |
| \@PackageError | 129 | \baselineskip | 928 |
| \@PackageInfo | 848, 853, 859, 866 | \begin | 852, 1023 |
| \@PackageWarningNoLine | 166 | \bfseries | 944 |
| \@ehc | 727 | \body | 655, 659 |
| \@empty | 854, 860, 867 | \boxA | 843, 857, 897, 898, 899, 900, 902, 905, 906, 907, 958, 959 |
| \@firstofone | 646, 649 | \boxB | 844, 864, 897, 899, 900, 903, 906, 959, 969 |
| \@gobble | 485, 643, 651 | \boxC | 845, 878, 898, 905, 907 |
| \@ifDefined | 723, 732, 739, 740 | \break | 226, 246, 248, 311, 331, 333 |
| \@ifdefinable | 731 | C | |
| \@ifundefined | 724 | \c | 995, 1053 |
| \@latex@error | 725 | \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, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701 |
| \@ne | 196, 198, 200, 202, 281, 283, 285, 287, 886, 1006 | \cdot | 793 |
| \@nil | 385, 388, 410, 414 | \char | 771 |
| \@onelevel@sanitize | <u>134</u> , 362, 436, 452 | \chardef | 392, 419, 423, 427, 429 |
| \@undefined | 110 | \clearpage | 981 |
| \[| 697 | \copy | 920, 958, 969 |
| \] | 223, 308, 693, 984, 1045 | \count@ | 640, 669, 673, 675, 676, 680, 682, 683, 684, 999, 1001, 1002, 1006 |
| \{ | 635, 694, 930 | \countdef | 640 |
| \} | 636, 695, 929 | \csname | 8, 21, 45, 58, 65, 111, 134, 361, 362, 639, 642, 645, 648, 706, 717, 726, 733, 734, 741, 742, 849, 1009 |
| \] | 698 | A | |
| \~ | 1002 | \abc | 991, 1056, 1062 |
| _ | 700, 771 | \abcx | 1055, 1058, 1059 |
| A | | \active | 103 |
| \abc | 991, 1056, 1062 | \advance | 196, 198, 200, 202, 281, 283, 285, 287, 676, 684, 1006 |
| \abx | 1055, 1058, 1059 | | |
| \active | 103 | | |
| \advance | 196, 198, 200, 202, 281, 283, 285, 287, 676, 684, 1006 | | |

| | | | | |
|---|---|---|--|---|
| <code>\CurrentOption</code> | 143 | | K | |
| | | D | <code>\kern</code> | 793 |
| <code>\DeclareOption</code> | 114, 119, 143 | | L | |
| <code>\DeclareRobustCommand</code> | 112, 117, 769, 780, 805 | | <code>\lccode</code> | 1002 |
| <code>\DeclareUnicodeCharacter</code> | 759 | | <code>\leavevmode</code> | 214, 299 |
| <code>\discretionary</code> | 789, 799 | | <code>\lefthyphenmin</code> | 783 |
| <code>\documentclass</code> | 110, 712 | | <code>\linebreak</code> | 249, 251, 334, 336, 771, 773, 983, 1046 |
| <code>\dp</code> | 896, 906, 907 | | <code>\llap</code> | 929 |
| | | E | <code>\LogTests</code> | 842 |
| <code>\empty</code> | 12 | | <code>\loop</code> | 654, 670, 681, 1000 |
| <code>\end</code> | 707, 908, 1064 | | <code>\lowercase</code> | 1003 |
| <code>\endcsname</code> | 8, 21, 45, 58, 65, 111, 134, 361, 362, 639, 642, 645, 648, 706, 717, 726, 733, 734, 741, 742, 849, 1009 | | M | |
| <code>\endinginput</code> | 30 | | <code>\makeatletter</code> | 722, 768, 979 |
| <code>\etex@unexpanded</code> | 445, 467, 475, 557, 582, 607, 626 | | <code>\makeatother</code> | 746, 1022 |
| <code>\Expect</code> | 887, 888, 889, 895, 896, 897, 898, 899, 905, 906, 907 | | <code>\maxdimen</code> | 884, 885 |
| | | G | <code>\mbox</code> | 236, 321, 958, 969, 988, 1041, 1052 |
| <code>\gdef</code> | 807, 835 | | <code>\meaning</code> | 137, 383, 407 |
| | | H | <code>\MessageBreak</code> | 167, 168, 169 |
| <code>\hbox</code> .. | 238, 323, 569, 581, 582, 594, 606, 607, 793, 798, 855, 862, 876 | | N | |
| <code>\hfil</code> | 226, 311 | | <code>\NeedsTeXFormat</code> | 711 |
| <code>\hfill</code> | 921, 961, 970 | | <code>\newcommand</code> | 113, 118, 723, 730, 738, 833, 846, 910, 917, 955, 980 |
| <code>\hfuzz</code> | 913 | | <code>\newsavebox</code> | 843, 844, 845 |
| <code>\hsize</code> | 912 | | <code>\next</code> | 660, 662, 664 |
| <code>\HstringA</code> | 856, 887, 888 | | <code>\nobreak</code> | 226, 231, 311, 316, 440 |
| <code>\HstringB</code> | 863, 887 | | <code>\nobreakspace</code> | 759 |
| <code>\HstringC</code> | 877, 888 | | <code>\noindent</code> | 919, 957, 968 |
| <code>\ht</code> | 895, 899, 900, 905, 959 | | <code>\null</code> | 211, 217, 225, 230, 296, 302, 310, 315, 439 |
| | | I | O | |
| <code>\if</code> | 178, 215, 263, 300 | | <code>\origshow</code> | 762, 765 |
| <code>\ifcase</code> | 9, 343, 346, 488, 496, 505, 513, 522, 530 | | P | |
| <code>\ifcat</code> | 252, 254, 337, 339 | | <code>\PackageError</code> | 115, 120, 129 |
| <code>\ifdim</code> | 890, 900, 924, 925, 959 | | <code>\PackageInfo</code> | 26 |
| <code>\ifetex@unexpanded</code> | 149 | | <code>\par</code> | 213, 214, 298, 299, 918, 953, 956, 967, 977 |
| <code>\ifnum</code> .. | 182, 184, 186, 188, 267, 269, 271, 273, 675, 683, 927, 935, 1001 | | <code>\PassOptionsToPackage</code> | 143 |
| <code>\ifSOUL@ignorespaces</code> | 218, 303 | | <code>\phantom</code> .. | 931, 937, 946, 951, 965, 974 |
| <code>\ifx</code> | 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 | | <code>\ProcessOptions</code> | 144 |
| <code>\immediate</code> | 23, 47 | | <code>\protect</code> .. | 540, 552, 565, 577, 590, 602 |
| <code>\IncludeTests</code> | 841 | | <code>\ProvidesPackage</code> | 59 |
| <code>\input</code> | 122, 125, 133, 702 | | Q | |
| <code>\iterate</code> | 656, 658, 660 | | <code>\qqquad</code> | 932, 938, 948 |
| | | R | | |
| | | <code>\raisebox</code> | | 928 |
| | | <code>\RangeCatcodeInvalid</code> | | 679, 688, 689, 690, 691 |
| | | <code>\repeat</code> | | 654, 666, 677, 685, 1007 |
| | | <code>\RequirePackage</code> | | 145, 146, 147 |
| | | <code>\RestoreCatcodes</code> .. | | 668, 671, 672, 703 |
| | | <code>\RestoreOrg</code> | | 738, 861, 868, 869, 870, 871, 872, 873, 874, 875 |
| | | <code>\righthyphenmin</code> | | 784 |
| | | <code>\rlap</code> | | 930, 936, 945, 950, 964, 973 |

S

| | |
|---|---|
| <code>\SaveOrg</code> | 730, 747, 748, 749, 750, 751, 752, 753, 754, 755 |
| <code>\section</code> | 1010 |
| <code>\setbox</code> | 855, 862, 876, 911 |
| <code>\SetTestVBox</code> | 857, 864, 878, <u>910</u> |
| <code>\show</code> | 762, 765 |
| <code>\showbox</code> | 892, 893, 902, 903 |
| <code>\showboxbreadth</code> | 885 |
| <code>\showboxdepth</code> | 884 |
| <code>\slash</code> | 232, 317, 1044 |
| <code>\small</code> | 923, 963, 972 |
| <code>\SOUL@</code> | 778, 803, 831 |
| <code>\SOUL@@</code> | 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 |
| <code>\SOuL@addfouroctets</code> | 354, <u>473</u> |
| <code>\SOUL@addgroup</code> | 253, 338, <u>538</u> |
| <code>\SOUL@addmath</code> | 255, 340, <u>563</u> |
| <code>\SOUL@addprotect</code> | 237, 239, 322, 324, <u>588</u> |
| <code>\SOuL@addthreectets</code> | 352, <u>450</u> |
| <code>\SOUL@addtoken</code> | . 444, 445, 466, 467, 474, 475, <u>613</u> |
| <code>\SOuL@addtwooctets</code> | 350, <u>434</u> |
| <code>\SOuL@analyzeutfviii</code> | 342, <u>379</u> |
| <code>\SOuL@analyzeutfviiix</code> | 344, <u>403</u> |
| <code>\SOUL@apo</code> | 186, 200, 271, 285, 505, 511, 513, 519 |
| <code>\SOuL@AtEnd</code> | 78, 79, 632 |
| <code>\SOuL@charhash</code> | 375, 408, 413 |
| <code>\SOUL@charkern</code> | 800 |
| <code>\SOuL@charthree</code> | 377, 422 |
| <code>\SOuL@chartwo</code> | 376, 426 |
| <code>\SOuL@checkutfviii</code> | 382, 387 |
| <code>\SOuL@checkutfviiix</code> | 406, <u>412</u> |
| <code>\SOuL@cmd</code> | 158, 163, 165, 174 |
| <code>\SOuL@cmdcheck</code> | 159, <u>162</u> |
| <code>\SOUL@comma</code> | 184, 198, 269, 283, 488, 494, 496, 502 |
| <code>\SOuL@defsanitizedstring</code> | 360, 364, 365, 366, 367, 368, 369, 370, 371 |
| <code>\SOUL@dotoken</code> | 257, 347 |
| <code>\SOUL@doword</code> | . 210, 216, 224, 229, 233, 242, 247, 250, 295, 301, 309, 314, 318, 327, 332, 335, 438, 454, 460 |
| <code>\SOuL@empty</code> | 378, 418 |
| <code>\SOUL@error</code> | <u>125</u> |
| <code>\SOUL@eval</code> | <u>176</u> |
| <code>\SOUL@eventuallyexhyphen</code> | . 211, 217, 225, 230, 234, 296, 302, 310, 315, 319, 439, 455, 461 |
| <code>\SOUL@everyexhyphen</code> ... | 774, 796, 819 |
| <code>\SOUL@everyhyphen</code> | 773, 788, 816 |
| <code>\SOUL@everylowerthan</code> | 777, 828 |
| <code>\SOUL@everyospace</code> | 220, 226, 231, 251, 305, 311, 316, 336, 440, 771, 786, 810 |
| <code>\SOUL@everysyllable</code> ... | 775, 787, 822 |
| <code>\SOUL@everytoken</code> | 776, 825 |
| <code>\SOUL@exhyphen</code> | 235, 320, 456, 462 |
| <code>\SOUL@flushapo</code> | . 187, 193, 206, 272, 278, 291, <u>504</u> |
| <code>\SOUL@flushcomma</code> | . 185, 192, 205, 270, 277, 290, <u>487</u> |
| <code>\SOUL@flushgrave</code> | . 189, 194, 207, 274, 279, 292, <u>521</u> |
| <code>\SOUL@flushminus</code> | 183, 191, 204, 268, 276, 289 |
| <code>\SOUL@grave</code> | 188, 202, 273, 287, 522, 528, 530, 536 |
| <code>\SOUL@hyphkern</code> | 790, 797 |
| <code>\SOUL@ignorespacesfalse</code> ... | 180, 265 |
| <code>\SOUL@ignorepacestrue</code> | 227, 312 |
| <code>\SOUL@minus</code> | 182, 196, 267, 281 |
| <code>\SOUL@n</code> | 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 |
| <code>\SOuL@next</code> | . 441, 443, 448, 457, 463, 465, 471 |
| <code>\SOuL@octets</code> | . 343, 346, 392, 419, 423, 427, 429 |
| <code>\SOuL@orgDeclareOption</code> | 114, 119 |
| <code>\SOuL@orgDeclareRobustCommand</code> .. | 112, 117 |
| <code>\SOuL@orgnewcommand</code> | 113, 118 |
| <code>\SOuL@orgPackageError</code> | 115, 120 |
| <code>\SOuL@orgSOUL@error</code> | 126, 130 |
| <code>\SOUL@postamble</code> | 772, 813 |
| <code>\SOUL@preamble</code> | 770, 782, 806 |
| <code>\SOuL@redefine</code> | <u>156</u> , 176, 477, 487, 504, 521, 538, 563, 588, 613 |
| <code>\SOuL@remainingtoken</code> | 482, <u>484</u> |
| <code>\SOuL@restorelatexcmds</code> | 116, 123 |
| <code>\SOUL@scan</code> | . 177, 214, 222, 244, 262, 299, 307, 329, 441, 457, 463, 549, 561, 574, 586, 599, 611, 621, 630 |
| <code>\SOUL@sethyphenchar</code> | 791 |
| <code>\SOUL@setkern</code> | 790, 797, 800 |
| <code>\SOUL@setup</code> | 781 |
| <code>\SOUL@spaceskip</code> | 243, 328 |
| <code>\SOUL@spc</code> | 178, 215, 263, 300 |
| <code>\SOUL@splittoken</code> | <u>477</u> |
| <code>\SOUL@stop</code> | 208, 293, 477, 484 |
| <code>\SOuL@stringcsnameu</code> | 409, 414 |
| <code>\SOuL@stringemdash</code> | 373, 459 |
| <code>\SOuL@stringendash</code> | 372, 453 |
| <code>\SOuL@stringfour</code> | 397 |
| <code>\SOuL@stringmacrocolon</code> | 408, 413 |
| <code>\SOuL@stringnobreakspace</code> ... | 374, 437 |
| <code>\SOuL@stringoctets</code> | 384, 388 |
| <code>\SOuL@stringthree</code> | 395 |
| <code>\SOuL@stringtwo</code> | 393 |
| <code>\SOuL@stringundeferr</code> | 409, 414 |
| <code>\SOuL@stringUTFviii</code> | 384, 388 |

| | | | |
|--|--|--|---|
| <code>\SOUL@syllable</code> | 787 | <code>\texttt</code> | 771, 992, 1049 |
| <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, 776, 787, 889, 895, 896, 897, 898, 899, 905, 906, 907, 930, 936, 945 |
| <code>\SOUL@token</code> | 478, 481, 776 | <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> | 807 | <code>\tracingonline</code> | 886 |
| <code>\soulH</code> | 805 | <code>\tw@</code> | 184, 186, 188, 269, 271, 273, 394, 429 |
| <code>\soulHadd</code> | 808, 811, 814, 817, 820, 823, 826, 829, 833 | | |
| <code>\soulHstring</code> | 835, 836, 854, 856, 860, 863, 867, 877 | | |
| <code>\soulomit</code> | 240, 241, 325, 326, 985, 1048 | | |
| <code>\space</code> | 786, 930, 958, 969 | | |
| <code>\spaceskip</code> | 243, 328 | | |
| <code>\ss</code> | 1029, 1032 | | |
| <code>\strip@prefix</code> | 137, 140 | | |
| <code>\subsection</code> | 847 | | |
| <code>\sy</code> | 780 | | |
| | | | |
| | T | | U |
| <code>\Test</code> | 687, 705, 979, 1024, 1025, 1026, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1062 | <code>\UNDEFINED</code> | 710, 1055 |
| <code>\TestDriver</code> | 846, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020 | <code>\unexpanded</code> | 710 |
| <code>\testdriver</code> | 849, 855, 862, 876, 914 | <code>\unskip</code> | 226, 311 |
| <code>\TestInfo</code> | 880, 881, 882, 917 | <code>\usepackage</code> | 713, 714, 715, 716, 718, 719, 720, 721, 756, 758, 763, 840 |
| <code>\TestInfoVBox</code> | 883, 955 | | |
| <code>\teststringA</code> | 850, 855, 857 | | |
| <code>\teststringB</code> | 851, 862, 864, 876, 878 | | |
| <code>\textbackslash</code> | 983, 984, 986, 989, 991, 993, 996 | | |
| <code>\textbraceleft</code> | 986, 989, 993, 996 | | |
| <code>\textbraceright</code> | 986, 989, 993, 996 | | |
| <code>\texteuro</code> | 1031 | | |
| | | | |
| | | | V |
| | | <code>\v</code> | 1036 |
| | | <code>\vbox</code> | 911 |
| | | <code>\vrule</code> | 958, 969 |
| | | | |
| | | | W |
| | | <code>\wd</code> | 889, 890, 897, 898, 924, 925, 930, 936, 945 |
| | | <code>\write</code> | 23, 47 |
| | | | |
| | | | X |
| | | <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, 1009, 1012 |
| | | | |
| | | | Z |
| | | <code>\z@</code> | 400, 419, 494, 502, 511, 519, 528, 536, 783, 784 |