

# The `showexpl` package\*

Rolf Niepraschk (Rolf.Niepraschk@ptb.de)

2006/09/18

## 1 Introduction

The documentation of a  $\LaTeX$  package is by far more readable if there are examples of the commands' and environments' usage. The best way to do that is to give a comparison of the  $\LaTeX$  code and the formatted output. `showexpl` is a package for doing that comparison, it is based on the package `listings` which provides a good typesetted source code with emphasised keywords and so on.

## 2 Usage

You can use `showexpl` like every other package by putting the line

```
\usepackage{showexpl}
```

in your source code. `showexpl` doesn't know any options by itself, but all options for the underlying packages (`listings` and `graphicx`) will be passed to the respective packages.

`showexpl` provides one command and one environment:

- `\LTXinputExample` and
- `LTXexample`

`\LTXinputExample` The syntax of `\LTXinputExample` is given by

```
\LTXinputExample[<key val list>]{<file>}
```

`LTXexample` The syntax of the environment `LTXexample` is given by

```
\begin{LTXexample}[<key val list>]\end{LTXexample}
```

The set of options represented by *<key val list>* is the same for both the command and the environment, the options are described in the following:

**attachfile** Boolean valued key, default value: false. If set to true the sourcecode will be attached to the `.pdf` file—presumed that the document is processed by `pdflatex`.

**codefile** Name of the (temporary) file that contains the code which will be formatted as source code. The default value is `\jobname.tmp`.

---

\*This document corresponds to `showexpl` v0.3g, dated 2006/09/18.

- explpreset** A *⟨key val list⟩* which serves for presetting the properties of the formatting of the source code, for values see the documentation of the `listings` package. The default value is
- graphic** Name of a (graphic) file. This file—if present—will be included and displayed instead of the formatted code. The default value is empty.
- hsep** Defines the horizontal distance between the source code and the formatted text.
- justification** Defines the justification of the formatted text: reasonable values are `\raggedleft`, `\raggedright`, `\centering`. The default value is `\raggedright`.
- overhang** A *dimen*-value that defines the amount by which the formatted text and the source code can overlap the print space. The default value is 0 pt.
- pos:** Defines the relative position of the formatted text relating to the source code. Allowed values are `a`, `b`, `l`, `r`, `o`, and `i` for above, below, left, right, outer, and inner. The last values give sense only for two-sided printing, where there are outer and inner margins of a page. The default value is `l`.
- preset** Any TeX code executed before the sample code but not visible in the listings area.
- rangeaccept** Boolean valued key, default value is false. If set to true, one can define ranges of lines that will be excerpted from the source code.
- rframe** Defines the form of the frame around the formatted text. With a non-empty value (e. g. “single”) a simple frame will be drawn. In the future more kinds of frames will be supported. The default value is empty (no frame).
- varwidth** Boolean valued key, default value is false. If set to true, the formatted text is set with its “natural” width instead of a fixed width as given by the value of the option `width`.
- hsep** Defines the vertical distance between the source code and the formatted text.
- wide** Boolean valued key, default value is false. If set to true, the source code and the formatted text overlap the print space and the margin area.
- width** A *⟨dimen⟩* value that defines the width of the formatted text. The default value depends of the relative positions of the source code and the formatted text.

### 3 Implementation

```

1 \DeclareOption{final}{%
2   \PassOptionsToPackage{\CurrentOption}{graphicx}%
3   \PassOptionsToPackage{\CurrentOption}{listings}%
4 }%
5 \DeclareOption{draft}{%
6   \PassOptionsToPackage{\CurrentOption}{graphicx}%

```

```

7 \PassOptionsToPackage{\CurrentOption}{listings}%
8 }%
9 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{listings}}
10 \ProcessOptions\relax
11 \RequirePackage{listings,calc,ifthen,graphicx,varwidth}
12 \AtEndOfPackage{\IfFileExists{attachfile.sty}%
13 {\RequirePackage{attachfile}}{\def\SX@attachfile{}}}

```

We must activate code from package listings for writing files.

```
14 \lst@RequireAspects{writefile}
```

`\SX@defaultWD` Parameter #2 is a length or a number. Parameter #1 is a macro. After a call of `\SX@defaultWD` this macro contains the value of the length or the value of the number multiplied by `\linewidth`.

```

15 \newcommand*\SX@defaultWD[2]{%
16 \afterassignment\SX@def@WD\dimen@#2\linewidth\relax{#1}}
17 \newcommand*\SX@def@WD{}
18 \def\SX@def@WD#1\relax#2{\edef#2{\the\dimen@}}

```

Additional keys.

```

19 \lst@Key{pos}\relax{\def\SX@pos{#1}}
20 \lst@Key{width}\relax{\def\SX@width{#1}}
21 \lst@Key{hsep}\relax{\@tempdima=#1\relax\edef\SX@hsep{\the\@tempdima}}
22 \lst@Key{vsep}\relax{\@tempdima=#1\relax\edef\SX@vsep{\the\@tempdima}}
23 \lst@Key{overhang}\relax{\def\SX@overhang{#1}}
24 \lst@Key{wide}f[t]{\lstKV@SetIf{#1}\if@SX@wide}
25 \lst@Key{rframe}\relax{\def\SX@rframe{#1}}
26 \lst@Key{preset}\relax{\def\SX@preset{#1}}

```

```

27 \lst@Key{explpreset}\relax{\def\SX@explpreset{#1}}
28 \lst@Key{codefile}\relax{\def\SX@codefile{#1}}
29 \newif\if@SX@rangeaccept \@SX@rangeacceptfalse
30 \newif\if@SX@varwidth \@SX@varwidthfalse
31 \newif\if@SX@wide \@SX@widefalse
32 \newif\if@SX@attachfile \@SX@attachfilefalse

```

```
33 \lst@Key{rangeaccept}f[t]{\lstKV@SetIf{#1}\if@SX@rangeaccept}
```

```

34 \lst@Key{varwidth}f[t]{\lstKV@SetIf{#1}\if@SX@varwidth}
35 \lst@Key{justification}\relax{\def\SX@justification{#1}}
36 \lst@Key{attachfile}f[t]{\lstKV@SetIf{#1}\if@SX@attachfile}
37 \newcommand*\SX@graphicname{}%
38 \newcommand*\SX@graphicparam{}%
39 \lst@Key{graphic}{ }[]{}%
40 \lstKV@OptArg[width=\linewidth]{#1}{}%
41 \edef\SX@graphicparam{##1}\edef\SX@graphicname{##2}%
42 }%
43 }%

```

```

44 \newbox\SX@ResBox
45 \newcommand*\SX@pos{}
46 \newcommand*\SX@width{}
47 \newcommand*\SX@hsep{}
48 \newcommand*\SX@vsep{}
49 \newcommand*\SX@overhang{}
50 \newcommand*\SX@rframe{}

```

```

51 \newcommand\SX@preset{}
52 \newcommand*\SX@explpreset{}

53 \newcommand*\SX@@explpreset{}
54 \newcommand*\SX@codefile{}\edef\SX@codefile{\jobname.tmp}
55 \newcommand*\SX@justification{\raggedright}

```

`\SX@@preset` Contains some redefinitions of L<sup>A</sup>T<sub>E</sub>X macros and environments to do nothing. `\SX@@preset` will be called just before typesetting the result of the example code. More can be added with the user key “`preset=...`”.

```

56 \newcommand*\SX@@preset{%
57   \renewcommand\documentclass[2] [] {\SX@eat@version}%
58   \renewcommand\usepackage[2] [] {\SX@eat@version}%
59   \renewenvironment{document}{-}{-}%
60   \renewenvironment{figure}[1] [] {\def\@capttype{figure}}{-%}
61   \renewenvironment{table}[1] [] {\def\@capttype{table}}{-%}
62   \renewcommand\cite[1] [] {}%
63   \let\tableofcontents\relax \let\listoffigures\relax
64   \let\listoftables\relax \let\printindex\relax
65   \let\listfiles\relax \let\nofiles\relax
66   \let\index@gobble \let\label@gobble
67   \let\bibliography@gobble
68   \let\pagestyle@gobble \let\thispagestyle@gobble
69   %%\let\immediate\relax \let\write@gobbletwo
70   %%\let\closeout@gobble \let\@input@gobble
71   \renewcommand\marginpar[2] [] {}%
72   \renewcommand\footnote[2] [] {}%
73   \let\@footnotetext@gobble
74   %%\abovedisplayskip=\z@
75   %%\abovedisplayshortskip=\z@
76 }
77 \newcommand*\SX@eat@version[1] [] {}

```

`\isSX@odd` Parameter #1 is executed on odd pages, parameter #2 on even pages.

```

78 \newif\ifSX@wasodd
79 \if@twoside
80   \newcommand*\isSX@odd[2] {%
81     \ifthenelse{\isodd{\pageref{\SX@IDENT}}}{%
82       {\SX@wasoddtrue #1}{\SX@wasoddfalse #2}}
83   \else
84     \newcommand*\isSX@odd[2] {#1}\SX@wasoddtrue
85   \fi

```

The call of `\isSX@odd` sets also `\ifSX@wasodd` to true or false. If it's clear that no page break occurs, `\ifSX@wasodd` can be used.

```

86 \newcounter{ltxexample}
87 \newcommand*\SX@IDENT{\SX@number\value{ltxexample}}

```

`\SX@attachfile`

```

88 \newcommand*\SX@attachfile{%
89   \if@SX@attachfile
90     \attachfile[mimetype=text/plain,subject={example \theltxexample}]%
91     {\SX@codefile}{-%}
92   \fi
93 }

```

`\SX@put@t/b/l/r/o/i` Six macros for positioning #2 (result) and #3 (code). The result can be above, below, left or right of the code area or on the outer or iinner side. Parameter #1 is the width of the result.

```

94 \newcommand*\SX@put@t[3]{%
95   \SX@ResultArea{\linewidth}{#2}\endgraf\pagebreak[2]%
96   \setlength\@tempdima{\SX@vsep}\vskip\@tempdima
97   \SX@CodeArea{\linewidth}{#3}%
98 }
99 \newcommand*\SX@put@b[3]{%
100  \SX@CodeArea{\linewidth}{#3}\endgraf\pagebreak[2]%
101  \setlength\@tempdima{\SX@vsep}\vskip\@tempdima
102  \SX@ResultArea{\linewidth}{#2}%
103 }
104 \newcommand*\SX@put@l[3]{%
105  \setlength\@tempdimc{\linewidth-#1-\SX@hsep}%
106  \SX@ResultArea{#1}{#2}\hfill\SX@CodeArea{\@tempdimc}{#3}%
107 }
108 \newcommand*\SX@put@r[3]{%
109  \setlength\@tempdimc{\linewidth-#1-\SX@hsep}%
110  \SX@CodeArea{\@tempdimc}{#3}\hfill\SX@ResultArea{#1}{#2}%
111 }
112 \newcommand*\SX@put@o[3]{%
113  \@nameuse{SX@put@\ifSX@wasodd r\else l\fi}{#1}{#2}{#3}%
114 }
115 \newcommand*\SX@put@i[3]{%
116  \@nameuse{SX@put@\ifSX@wasodd l\else r\fi}{#1}{#2}{#3}%
117 }
118 \newcommand\SX@ResultArea[2]{%
119  \SX@justification\setlength\@tempdima{#1}%
120  %\minipage\@tempdima#2\endminipage
121  \parbox\@tempdima{#2}%
122 }
123 \newcommand\SX@CodeArea[2]{%
124  \setlength\@tempdima{#1}%
125  \sbox\@tempboxa{\parbox\@tempdima{#2}}%
126  \@tempdima=\dp\@tempboxa\usebox\@tempboxa
127  \rlap{\raisebox{-\@tempdima}[Opt][Opt]{\SX@attachfile}}%
128 }
129 \newcommand*\SX@KillAboveCaptionskip{%
130  \ifx\lst@caption\@empty\else
131    \lst@ifsubstring t\lst@captionpos
132      {\vskip-\abovecaptionskip}{}%
133  \fi
134 }
135 \newcommand*\SX@KillBelowCaptionskip{%
136  \ifx\lst@caption\@empty\else
137    \lst@ifsubstring b\lst@captionpos
138      {\vskip-\belowcaptionskip}{}%
139  \fi
140 }

```

LTXexample

```

141 \lstnewenvironment{LTXexample}[1] []
142 {%

```

```

143 \temptokena{#1}%
144 \begingroup
    For "codefile=..."/"graphic=..." if \theltxexample or \thelstlisting is part of
    the filename.
145 \advance\c@ltxexample\@ne \advance\c@lstlisting\@ne
146 \expandafter\lstset\expandafter{\SX@explpreset,#1}%
147 \edef\x{\endgroup
148 \def\noexpand\SX@codefile{\SX@codefile}%
149 \def\noexpand\SX@graphicname{\SX@graphicname}%
150 \def\noexpand\SX@graphicparam{\SX@graphicparam}}%
151 \x
152 \xdef\SX@@explpreset{\the\temptokena,codefile=\SX@codefile,
153 graphic={[\SX@graphicparam]{\SX@graphicname}}}%
154 \setbox\@tempboxa=\hbox\bgroup% Warum noetig?
155 \lst@BeginWriteFile{\SX@codefile}%
156 }
157 {%
158 \lst@EndWriteFile\egroup
159 \SX@put@code@result
160 }

161 \newcommand*\SX@put@code@result{%
162 \begingroup
163 \expandafter\lstset\expandafter{\SX@explpreset}%
164 \let\lst@float=\relax\let\SX@float=\relax

    Without the following call \lst@beginfloat is undefined.
165 \expandafter\lstset\expandafter{\SX@@explpreset}%
166 \ifx\lst@float\relax\else

    \lst@float must be \relax because the whole "example" should float but not
    the listings part in addition.
167 \let\SX@float=\lst@float\let\lst@float=\relax
168 \g@addto@macro\SX@@explpreset{,float=false}%
169 \edef\@tempa{\noexpand\lst@beginfloat{lstlisting}[\SX@float]}%
170 \expandafter\@tempa
171 \fi
172 \ifx\lst@caption\@empty
173 \lstset{no1ol=true}%
174 \fi
175 \if\SX@wide\def\SX@overhang{\marginparwidth+\marginparsep}\fi
176 \trivlist\item\relax
177 \stepcounter{ltxexample}\label{\SX@IDENT}%

    Make \SX@width a real dimension if the unit is missing.
178 \SX@defaultWD\SX@width{\SX@width}%

    Set the default width if necessary.
179 \ifdim\SX@width<\z@
180 \@tempwattrue
181 \def\@tempa{t}%
182 \ifx\@tempa\SX@pos\@tempwafalse\fi
183 \def\@tempa{b}%
184 \ifx\@tempa\SX@pos\@tempwafalse\fi

```

```

185     \setlength\@tempdima{\linewidth+\SX@overhang}%
186     \if@tempswa\@tempdima=.5\@tempdima\fi%
187     \edef\SX@width{\the\@tempdima}%
188     \fi
Correct \SX@width if a frame is requested.
189     \ifx\SX@rframe\@empty
190     \long\def\SX@frame##1{##1}%
191     \else
192     \let\SX@frame\fbbox
193     \setlength\@tempdima{\SX@width-2\fbboxsep-2\fbboxrule}%
194     \edef\SX@width{\the\@tempdima}%
195     \fi
196     \isSX@odd{\def\@tempa{1}}{\def\@tempa{r}}%
197     \makebox[\linewidth][\@tempa]{%
198     \parbox{\linewidth+\SX@overhang}{%
\SX@codefile (\jobname.tmp) is not necessary for the filelist.
199     \let\@addtofilelist\@gobble
200     \let\lst@ifdisplaystyle=\iftrue
201     \SX@KillAboveCaptionskip\lst@MakeCaption{t}%
202     \lst@belowskip=\z@
203     \let\SX@MakeCaption\lst@MakeCaption
204     \let\lst@MakeCaption\@gobble
Use the “natural” width of the result code if “varwidth” is true.
205     \setbox\SX@ResBox\hbox{%
206     \SX@frame{%
207     \@nameuse{\if\SX@varwidth varwidth\else minipage\fi}%
208     \SX@width\relax\SX@resultInput%
209     \@nameuse{end\if\SX@varwidth varwidth\else minipage\fi}}%
210     \edef\SX@width{\the\wd\SX@ResBox}%
211     \@ifundefined{SX@put@\SX@pos}%
212     {\@latex@error{Parameter ‘\SX@pos’ undefined}\@ehd}%
213     {\@nameuse{SX@put@\SX@pos}}%
214     {\SX@width}{\box\SX@ResBox}{\SX@codeInput}}%
215     \let\lst@MakeCaption\SX@MakeCaption
216     \lst@MakeCaption{b}\SX@KillBelowCaptionskip
217     }%
218     }%
219     \endtrivlist
220     \ifx\SX@float\relax\else\expandafter\lst@endfloat\fi
221     \gdef\SX@@explpreset{}%
222     \endgroup
223 }

224 \newcommand\SX@SkipToFirst{%
225     \ifeof\@inputcheck\else
226     \ifnum \lst@lineno=\lst@firstline\else
227     \readline\@inputcheck to\SX@tempa
228     \typeout{IGNORE (\the\lst@lineno)}%
229     \global\advance\lst@lineno\@ne
230     \SX@SkipToFirst
231     \fi
232     \fi
233 }

```

```

234 \newcommand\SX@ProcessResult{%
235   \ifeof\@inputcheck
236     \let\SX@tempb\relax
237   \else
238     \let\SX@tempb\SX@ProcessResult
239     \ifnum \lst@lineno>\lst@lastline\relax
240       \ifx\lst@linerange\@empty
241         \let\SX@tempb\relax
242       \else
243         \lst@GetLineInterval
244         \SX@SkipToFirst
245       \fi
246     \else
247       \readline\@inputcheck to\SX@tempa
248       \typeout{READ (\the\lst@lineno)}%
249       \expandafter\g@addto@macro
250       \expandafter\SX@lines\expandafter{\SX@tempa^^J}%
251       \global\advance\lst@lineno\@ne
252     \fi
253   \fi
254   \SX@tempb
255 }

```

\SX@input

```

256 \newcommand\SX@input[1]{%
257   \begingroup
258     \IfFileExists{#1}{}%
259     {%
260       \filename@parse{#1}%
261       \ifx\filename@ext\relax \def\filename@ext{tex}\fi
262       \@latexerr{File
263         ‘\filename@area\filename@base.\filename@ext’ not found.^^J^^J}\@ehd%
264     }%
265     \openin\@inputcheck#1
266     \lsthk@PreSet\let\lst@linerange\@empty\global\lst@lineno\@ne
267     \expandafter\lstset\expandafter{\SX@@explpreset}%
268     \ifx\lst@linerange\@empty
269       \edef\lst@linerange{{\lst@firstline}-{\lst@lastline}},}%
270     \fi
271     \lst@GetLineInterval
272     \SX@Info
273     \newlinechar=‘^^J\relax
274     \SX@SkipToFirst\let\SX@lines\@empty
275     \SX@ProcessResult
276     \closein\@inputcheck
277     \scantokens\expandafter{\SX@lines}%
278   \endgroup
279 }

280 \newcommand*\SX@Info{%
281   \typeout{-----}%
282   \typeout{pos=\SX@pos}%
283   \typeout{width=\SX@width}%
284   \typeout{hsep=\SX@hsep}%

```

```

285 \typeout{vsep=\SX@vsep}%
286 \typeout{overhang=\SX@overhang}%
287 \typeout{rframe=\SX@rframe}%
288 \typeout{codefile=\SX@codefile}%
289 \@ifundefined{lst@firstline}{}%
290   {\typeout{\string\lst@firstline=\lst@firstline}}%
291 \@ifundefined{lst@lastline}{}%
292   {\typeout{\string\lst@lastline=\lst@lastline}}%
293 \@ifundefined{lst@linrange}{}%
294   {\typeout{\string\lst@linrange=\lst@linrange}}%
295 \typeout{\string\if@SX@wide=\if@SX@wide TRUE\else FALSE\fi}%
296 \typeout{\string\if@SX@rangeaccept=\if@SX@rangeaccept TRUE\else FALSE\fi}%
297 \typeout{\string\if@SX@varwidth=\if@SX@varwidth TRUE\else FALSE\fi}%
298 \typeout{graphicfile=\SX@graphicname, graphicparameter=[\SX@graphicparam]}%
299 \typeout{-----}%
300 }
301 \providecommand* \MakePercentIgnore{\catcode'\%9\relax}
302 \providecommand* \MakePercentComment{\catcode'\%14\relax}

```

\SX@resultInput

```

303 \newcommand*\SX@resultInput{%
304   \ifx\SX@graphicname\@empty
305     \begingroup
306       \MakePercentComment\makeatother\catcode'\^^M=5\relax
307       \SX@@preset\SX@preset
308       \if@SX@rangeaccept
309         \let\SX@tempa=\SX@input
310       \else
311         \let\SX@tempa=\input
312       \fi
313       \SX@tempa{\SX@codefile}\par%
314     \endgroup
315   \else
316     \expandafter\includegraphics\expandafter[\SX@graphicparam]%
317     {\SX@graphicname}%
318   \fi
319 }

```

\SX@codeInput

```

320 \newcommand*\SX@codeInput{%
  Without a caption entry the command \lstinputlisting adds the filename to
  the “list of listings” (lol). This should be avoided.
321   \begingroup
  The default parameters for all examples.
322   \expandafter\lstset\expandafter{\SX@explpreset}%
  If ”numbers=none” then margin dimensions should be zero.
323   \expandafter\lstset\expandafter{\SX@@explpreset}%
324   \ifx\lst@PlaceNumber\@empty
325     \g@addto@macro\SX@@explpreset{,xleftmargin=0pt,xrightmargin=0pt}%
326   \fi
327   \SX@Info
328   \expandafter\lstinputlisting\expandafter%

```

```

329     [\SX@@explpreset,nolol=true,caption={}] {\SX@codefile}%
330 \endgroup
331 }%

332 \newcommand*\LTXinputExample[2] [] {%
333   \g@addto@macro\SX@@explpreset{#1,codefile=#2}%
334   \SX@put@code@result}%

All the default values.
335 \lstset{explpreset={numbers=left,numberstyle=\tiny,numbersep=.3em,
Negative width means defaults.
336 xleftmargin=1em,columns=flexible,language=[LaTeX]TEX},pos=1,width=-99pt,
337 overhang=0pt,hsep=\columnsep,vsep=\bigskipamount,rframe=single}

Changing the defaults possible in showexpl.cfg.
338 \InputIfFileExists{showexpl.cfg}{-}{-}

```

## Change History

v0.1a	General: “hpos” and “vpos” added, “pos” removed (RN). . . . . 3	v0.1k	General: Some bug corrections (RN). . . . . 3
	Initial version . . . . . 1		\SX@put@t/b/l/r/o/i: Change [a]bove to [t]op (RN). . . . . 5
v0.1b	\SX@put@t/b/l/r/o/i: Positioning the captions more independend of the result and code area (RN). . . . . 5	v0.1l	General: “graphic” added (RN). . . 3
v0.1c	\SX@put@t/b/l/r/o/i: Commands \SX@KillAboveCaptionskip and \SX@KillBelowCaptionskip added (RN). . . . . 5	v0.1m	General: Problem related to \label/\ref solved (RN). . . . . 6
v0.1f	General: “lstpreset” added. (RN). 3	v0.2a	General: “varwidth” and “justifica- tion” added (RN). . . . . 3
v0.1h	General: “codefile” added. (RN). . 3		“varwidth” package used (RN). . 6
	“lstpreset” renamed to “explpre- set” (RN). . . . . 3	v0.2b	General: Check if \SX@put@? is de- fined (RN). . . . . 6
	New macro \LTXinputExample (RN). . . . . 10	v0.3a	General: “attachfile” added (RN). 3
	LTXexample: Renamed from “exam- ple” to “LTXexample’ (RN). . . 5		\SX@attachfile: Attach file func- tionality (with pdfTEX) added (RN). . . . . 4
v0.1i	General: Better caption positioning and correct distance between the parts (RN). . . . . 6	v0.3b	\SX@resultInput: Input of re- sult code now inside a group; \makeatother added (RN). . . . 9
v0.1j	General: “rangeaccept” added (RN). . . . . 3	v0.3c	\SX@resultInput: Wrong catcode for newline char corrected (RN). . . . . 9
	\SX@input: For ranges of lines (RN). . . . . 8	v0.3d	\SX@resultInput: Missing \par added (RN). . . . . 9

v0.3e		<code>\readline</code> and <code>\scantokens</code> .
<code>\SX@preset</code> : More redefinitions		Thanks to Ulrich Diez for help
added (RN). . . . .	4	(RN). . . . .
v0.3g		Missing <code>\newcommand</code> for
General: <code>\SX@ProcessResult</code> is		<code>\SX@explpreset</code> added (RN). 4
now working correctly using		

## 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.

	<b>Symbols</b>	<code>\closeout</code> . . . . . 70	<code>\isSX@odd</code> . . . . . <u>78</u> , 196
<code>\%</code> . . . . .	301, 302	<code>\columnsep</code> . . . . . 337	
<code>\@input</code> . . . . .	70		<b>L</b>
<code>\SX@attachfilefalse</code> 32		<b>E</b>	<code>\label</code> . . . . . 66, 177
<code>\SX@rangeacceptfalse</code>		<code>\endgraf</code> . . . . . 95, 100	<code>\listoffigures</code> . . . . 63
. . . . .	29	environments:	<code>\listoftables</code> . . . . . 64
<code>\SX@varwidthfalse</code> .	30	<code>LTXexample</code> . . . . . <u>141</u>	<code>\lst@beginfloat</code> . . . 169
<code>\SX@widefalse</code> . . . .	31		<code>\lst@BeginWriteFile</code> 155
<code>\@addtofilelist</code> . . .	199	<b>F</b>	<code>\lst@belowskip</code> . . . . 202
<code>\@capttype</code> . . . . .	60, 61	<code>\fbox</code> . . . . . 192	<code>\lst@caption</code> . . . . .
<code>\@ehd</code> . . . . .	212, 263	<code>\fboxrule</code> . . . . . 193	. . . . . 130, 136, 172
<code>\@footnotetext</code> . . . .	73	<code>\fboxsep</code> . . . . . 193	<code>\lst@captionpos</code> 131, 137
<code>\@gobble</code> . . . . .	66–	<code>\filename@area</code> . . . . 263	<code>\lst@endfloat</code> . . . . . 220
68, 70, 73, 199, 204		<code>\filename@base</code> . . . . 263	<code>\lst@EndWriteFile</code> . 158
<code>\@gobbletwo</code> . . . . .	69	<code>\filename@ext</code> . 261, 263	<code>\lst@firstline</code> . . . .
<code>\@inputcheck</code> 225, 227,		<code>\filename@parse</code> . . . 260	. . . . . 226, 269, 290
235, 247, 265, 276		<code>\footnote</code> . . . . . 72	<code>\lst@float</code> 164, 166, 167
<code>\@latex@error</code> . . . . .	212		<code>\lst@GetLineInterval</code>
<code>\@latexerr</code> . . . . .	262	<b>G</b>	. . . . . 243, 271
<code>\@temptokena</code> . . 143, 152		<code>\g@addto@macro</code> . . . .	<code>\lst@ifdisplaystyle</code> 200
<code>\^</code> . . . . .	273, 306	. 168, 249, 325, 333	<code>\lst@ifSubstring</code> . .
			. . . . . 131, 137
<b>A</b>		<b>I</b>	<code>\lst@Key</code> 19–28, 33–36, 39
<code>\abovecaptionskip</code> . 132		<code>\if@SX@attachfile</code> .	<code>\lst@lastline</code> . . . .
<code>\abovedisplayshortskip</code>		. . . . . 32, 36, 89	. . . . . 239, 269, 292
. . . . .	75	<code>\if@SX@rangeaccept</code> .	<code>\lst@lineno</code> . . . . .
<code>\abovedisplayskip</code> . 74		. . 29, 33, 296, 308	. 226, 228, 229,
<code>\attachfile</code> . . . . .	90	<code>\if@SX@varwidth</code> . . .	239, 248, 251, 266
		30, 34, 207, 209, 297	<code>\lst@linerange</code> 240,
<b>B</b>		<code>\if@SX@wide</code> . . . . .	266, 268, 269, 294
<code>\belowcaptionskip</code> . 138		. . 24, 31, 175, 295	<code>\lst@MakeCaption</code> 201,
<code>\bibliography</code> . . . . .	67	<code>\if@twoside</code> . . . . . 79	203, 204, 215, 216
<code>\bigskipamount</code> . . . .	337	<code>\ifeof</code> . . . . . 225, 235	<code>\lst@PlaceNumber</code> . . 324
<code>\box</code> . . . . .	214	<code>\IfFileExists</code> . . 12, 258	<code>\lst@RequireAspects</code> 14
		<code>\ifSX@wasodd</code> 78, 113, 116	<code>\lsthk@PreSet</code> . . . . . 266
<b>C</b>		<code>\ifthenelse</code> . . . . . 81	<code>\lstinputlisting</code> . . 328
<code>\c@lstlisting</code> . . . . .	145	<code>\immediate</code> . . . . . 69	<code>\lstKV@OptArg</code> . . . . . 40
<code>\c@ltxexample</code> . . . . .	145	<code>\includegraphics</code> . . 316	<code>\lstKV@SetIf</code> . . . . .
<code>\cite</code> . . . . .	62	<code>\index</code> . . . . . 66	. . . . . 24, 33, 34, 36
<code>\closein</code> . . . . .	276	<code>\isodd</code> . . . . . 81	<code>\lstnewenvironment</code> . 141

<code>\lstset</code> . . . . .	146, 163, 165, 173, 267, 322, 323, 335
<code>LTExample</code> (environ- ment) . . . . .	<u>141</u>
<code>\LTXinputExample</code> . .	332
<b>M</b>	
<code>\makeatother</code> . . . . .	306
<code>\makebox</code> . . . . .	197
<code>\MakePercentComment</code> . . . . .	302, 306
<code>\MakePercentIgnore</code> .	301
<code>\marginpar</code> . . . . .	71
<code>\marginparsep</code> . . . .	175
<code>\marginparwidth</code> . . .	175
<b>N</b>	
<code>\newbox</code> . . . . .	44
<code>\newlinechar</code> . . . . .	273
<b>O</b>	
<code>\openin</code> . . . . .	265
<b>P</b>	
<code>\pagebreak</code> . . . . .	95, 100
<code>\pageref</code> . . . . .	81
<code>\pagestyle</code> . . . . .	68
<code>\printindex</code> . . . . .	64
<b>R</b>	
<code>\raggedright</code> . . . . .	55
<code>\raisebox</code> . . . . .	127
<code>\readline</code> . . . . .	227, 247
<code>\rlap</code> . . . . .	127
<b>S</b>	
<code>\sbox</code> . . . . .	125
<code>\scantokens</code> . . . . .	277
<code>\stepcounter</code> . . . . .	177
<code>\string</code> 290, 292, 294–297	
<code>\SX@@explpreset</code> . . .	. . . 53, 152, 165, 168, 221, 267, 323, 325, 329, 333
<code>\SX@preset</code> . . . . .	<u>56</u> , 307
<code>\SX@attachfile</code> . . . .	. . . . . 13, <u>88</u> , 127
<code>\SX@CodeArea</code> . . . . .	97, 100, 106, 110, 123
<code>\SX@codefile</code> . . . . .	28, 54, 91, 148, 152, 155, 288, 313, 329
<code>\SX@codeInput</code> . . . . .	214, <u>320</u>
<code>\SX@def@WD</code> . . . . .	16–18
<code>\SX@defaultWD</code> . . . .	<u>15</u> , 178
<code>\SX@eat@version</code> . . . .	. . . . . 57, 58, 77
<code>\SX@explpreset</code> . . . . .	27, 52, 146, 163, 322
<code>\SX@float</code> . . . . .	. . . . . 164, 167, 169, 220
<code>\SX@frame</code> . . . . .	190, 192, 206
<code>\SX@graphicname</code> . . . .	. . . . . 37, 41, 149, 153, 298, 304, 317
<code>\SX@graphicparam</code> . . . .	. . . . . 38, 41, 150, 153, 298, 316
<code>\SX@hsep</code> . . . . .	. . . . . 21, 47, 105, 109, 284
<code>\SX@IDENT</code> . . . . .	81, 87, 177
<code>\SX@Info</code> . . . . .	272, 280, 327
<code>\SX@input</code> . . . . .	<u>256</u> , 309
<code>\SX@justification</code> . . . .	. . . . . 35, 55, 119
<code>\SX@KillAboveCaptionskip</code> . . . . .	129, 201
<code>\SX@KillBelowCaptionskip</code> . . . . .	135, 216
<code>\SX@lines</code> . . . . .	250, 274, 277
<code>\SX@MakeCaption</code> 203, 215	
<code>\SX@overhang</code> . . . . .	23, 49, 175, 185, 198, 286
<code>\SX@pos</code> . . . . .	19, 45, 182, 184, 211–213, 282
<code>\SX@preset</code> . . . . .	26, 51, 307
<code>\SX@ProcessResult</code> . . . .	. . . . . 234, 238, 275
<code>\SX@put@code@result</code> . . . . .	159, 161, 334
<code>\SX@put@t</code> . . . . .	94
<code>\SX@put@t/b/l/r/o/i</code> <u>94</u>	
<code>\SX@ResBox</code> . . . . .	. . . . . 44, 205, 210, 214
<code>\SX@ResultArea</code> . . . . .	95, 102, 106, 110, 118
<code>\SX@resultInput</code> 208, <u>303</u>	
<code>\SX@rframe</code> . . . . .	. . . . . 25, 50, 189, 287
<code>\SX@SkipToFirst</code> . . . . .	. . . . . 224, 230, 244, 274
<code>\SX@tempa</code> . . . . .	227, 247, 250, 309, 311, 313
<code>\SX@tempb</code> . . . . .	. . . . . 236, 238, 241, 254
<code>\SX@vsep</code> . . . . .	. . . . . 22, 48, 96, 101, 285
<code>\SX@wasoddfalse</code> . . . . .	82
<code>\SX@wasoddtrue</code> . . . . .	82, 84
<code>\SX@width</code> . . . . .	. . . . . 20, 46, 178, 179, 187, 193, 194, 208, 210, 214, 283
<b>T</b>	
<code>\theltxexample</code> . . . . .	90
<code>\thispagestyle</code> . . . . .	68
<b>U</b>	
<code>\usebox</code> . . . . .	126
<b>W</b>	
<code>\write</code> . . . . .	69