

The `showexpl` package*

Rolf Niepraschk (Rolf.Niepraschk@gmx.de)

2013/03/21

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 $\langle key\ val\ list \rangle$ 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` v0.3k, dated 2013/03/21.

- explpreset** A $\langle key\ val\ list \rangle$ 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 `t`, `b`, `l`, `r`, `o`, and `i` for top, bottom, 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 $\langle dimen \rangle$ 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{attachfiles}{%
10 \AtBeginDocument{\IfFileExists{attachfile.sty}%
11 {\RequirePackage{attachfile}}{\def\SX@attachfile{}}}
12 }%
13 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{listings}}
14 \ProcessOptions\relax
15 \RequirePackage{listings,calc,ifthen,graphicx,varwidth}

We must activate code from package listings for writing files.
16 \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.
17 \newcommand*\SX@defaultWD[2]{%
18 \afterassignment\SX@def@WD\dimen@#2\linewidth\relax{#1}}
19 \newcommand*\SX@def@WD{}
20 \def\SX@def@WD#1\relax#2{\edef#2{\the\dimen@}}

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

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

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

36 \lst@Key{varwidth}f[t]{\lstKV@SetIf{#1}\if@SX@varwidth}
37 \lst@Key{justification}\relax{\def\SX@justification{#1}}
38 \lst@Key{attachfile}f[t]{\lstKV@SetIf{#1}\if@SX@attachfile}
39 \newcommand*\SX@graphicname{}%
40 \newcommand*\SX@graphicparam{}%
41 \lst@Key{graphic}{ }[]{}%
42 \lstKV@OptArg[width=\linewidth]{#1}{%
43 \edef\SX@graphicparam{##1}\edef\SX@graphicname{##2}%
44 }%
45 }%
46 \newbox\SX@ResBox
47 \newcommand*\SX@pos{}
48 \newcommand*\SX@width{}
49 \newcommand*\SX@hsep{}
50 \newcommand*\SX@vsep{}

```

```

51 \newcommand*\SX@overhang{}
52 \newcommand*\SX@rframe{}
53 \newcommand*\SX@preset{}
54 \newcommand*\SX@explpreset{}

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

```

`\SX@@preset` Contains some redefinitions of L^AT_EX 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=...`”.

```

58 \newcommand*\SX@@preset{%
59   \renewcommand\documentclass[2] [] {\SX@eat@version}%
60   \renewcommand\usepackage[2] [] {\SX@eat@version}%
61   \renewenvironment{document}{}{}%
62   \renewenvironment{figure}[1] [] {\def\@capttype{figure}}{}%
63   \renewenvironment{table}[1] [] {\def\@capttype{table}}{}%
64   \renewcommand\cite[1] [] {}%
65   \let\tableofcontents\relax \let\listoffigures\relax
66   \let\listoftables\relax \let\printindex\relax
67   \let\listfiles\relax \let\nofiles\relax
68   \let\index\@gobble \let\label\@gobble
69   \let\bibliography\@gobble
70   \let\pagestyle\@gobble \let\thispagestyle\@gobble
71   %%\let\immediate\relax \let\write\@gobbletwo
72   %%\let\closeout\@gobble \let\@input\@gobble
73   \renewcommand\marginpar[2] [] {}%
74   \renewcommand\footnote[2] [] {}%
75   \let\@footnotetext\@gobble
76   %%\abovedisplayskip=\z@
77   %%\abovedisplayskip=\z@
78 }
79 \newcommand*\SX@eat@version[1] [] {}

```

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

```

80 \newif\ifSX@wasodd
81 \if@twoside
82   \newcommand*\isSX@odd[2] {%
83     \ifthenelse{\isodd{\pageref{\SX@IDENT}}}%
84       {\SX@wasoddtrue #1}{\SX@wasoddfalse #2}}
85 \else
86   \newcommand*\isSX@odd[2] {\#1}\SX@wasoddtrue
87 \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.

```

88 \newcounter{ltxexample}
89 \newcommand*\SX@IDENT{\SX@\number\value{ltxexample}}

```

`\SX@attachfile`

```

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

```

```

93      {\SX@codefile}{}%
94  \fi
95 }

```

\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 inner side. Parameter #1 is the width of the result.

```

96 \newcommand*\SX@put@t[3]{%
97   \SX@ResultArea{\linewidth}{#2}\endgraf\pagebreak[2]%
98   \setlength\@tempdima{\SX@vsep}\vskip\@tempdima
99   \SX@CodeArea{\linewidth}{#3}%
100 }
101 \newcommand*\SX@put@b[3]{%
102   \SX@CodeArea{\linewidth}{#3}\endgraf\pagebreak[2]%
103   \setlength\@tempdima{\SX@vsep}\vskip\@tempdima
104   \SX@ResultArea{\linewidth}{#2}%
105 }
106 \newcommand*\SX@put@l[3]{%
107   \setlength\@tempdimc{\linewidth-#1-\SX@hsep}%
108   \SX@ResultArea{#1}{#2}\hfill\SX@CodeArea{\@tempdimc}{#3}%
109 }
110 \newcommand*\SX@put@r[3]{%
111   \setlength\@tempdimc{\linewidth-#1-\SX@hsep}%
112   \SX@CodeArea{\@tempdimc}{#3}\hfill\SX@ResultArea{#1}{#2}%
113 }
114 \newcommand*\SX@put@o[3]{%
115   \@nameuse{SX@put@\ifSX@wasodd r\else l\fi}{#1}{#2}{#3}%
116 }
117 \newcommand*\SX@put@i[3]{%
118   \@nameuse{SX@put@\ifSX@wasodd l\else r\fi}{#1}{#2}{#3}%
119 }
120 \newcommand\SX@ResultArea[2]{%
121   \SX@justification\setlength\@tempdima{#1}%
122   %\minipage\@tempdima#2\endminipage
123   \parbox\@tempdima{#2}%
124 }
125 \newcommand\SX@CodeArea[2]{%
126   \setlength\@tempdima{#1}%
127   \sbox\@tempboxa{\parbox\@tempdima{#2}}%
128   \@tempdima=\dp\@tempboxa\usebox\@tempboxa
129   \rlap{\raisebox{-\@tempdima}[Opt][Opt]{\SX@attachfile}}%
130 }
131 \newcommand*\SX@KillAboveCaptionskip{%
132   \ifx\lst@caption\@empty\else
133     \lst@ifsubstring t\lst@captionpos
134     {\vskip-\abovecaptionskip}{}%
135   \fi
136 }
137 \newcommand*\SX@KillBelowCaptionskip{%
138   \ifx\lst@caption\@empty\else
139     \lst@ifsubstring b\lst@captionpos
140     {\vskip-\belowcaptionskip}{}%
141   \fi
142 }

```

LTXexample

```

143 \lstnewenvironment{LTXexample}[1] []
144 {%
145   \@temptokena{#1}%
146   \begingroup
    For "codefile=..." / "graphic=..." if \theltxexample or \thelstlisting is part of
    the filename.
147   \advance\c@ltxexample\@ne \advance\c@lstlisting\@ne
148   \expandafter\lstset\expandafter{\SX@explpreset,#1}%
149   \edef\x{\endgroup
150     \def\noexpand\SX@codefile{\SX@codefile}%
151     \def\noexpand\SX@graphicname{\SX@graphicname}%
152     \def\noexpand\SX@graphicparam{\SX@graphicparam}}%
153   \x
154   \xdef\SX@@explpreset{\the\@temptokena,codefile=\SX@codefile,
155     graphic={[\SX@graphicparam]{\SX@graphicname}}}%
156   \setbox\@tempboxa=\hbox\bgroup% Warum noetig?
157   \lst@BeginWriteFile{\SX@codefile}%
158 }
159 {%
160   \lst@EndWriteFile\egroup
161   \SX@put@code@result
162 }

```

\SX@put@code@result

```

163 \newcommand*\SX@put@code@result{%
164   \begingroup
165   \expandafter\lstset\expandafter{\SX@explpreset}%
166   \let\lst@float=\relax\let\SX@float=\relax
    Without the following call \lst@beginfloat is undefined.
167   \expandafter\lstset\expandafter{\SX@@explpreset}%
168   \ifx\lst@float\relax\else
    \lst@float must be \relax because the whole "example" should float but not
    the listings part in addition.
169     \let\SX@float=\lst@float\let\lst@float=\relax
170     \g@addto@macro\SX@@explpreset{,float=false}%
171     \edef\@tempa{\noexpand\lst@beginfloat{lstlisting}[\SX@float]}%
172     \expandafter\@tempa
173   \fi
174   \ifx\lst@caption\@empty
175     \lstset{nolol=true}%
176   \fi
177   \if\SX@wide\def\SX@overhang{\marginparwidth+\marginparsep}\fi
178   \trivlist\item\relax
179   \stepcounter{ltxexample}\label{\SX@IDENT}%
    Make \SX@width a real dimension if the unit is missing.
180   \SX@defaultWD\SX@width{\SX@width}%
    Set the default width if necessary.
181   \ifdim\SX@width<\z@
182     \@tempwattrue

```

```

183     \def\@tempa{t}%
184     \ifx\@tempa\SX@pos\@tempswafalse\fi
185     \def\@tempa{b}%
186     \ifx\@tempa\SX@pos\@tempswafalse\fi
187     \setlength\@tempdima{\linewidth+\SX@overhang}%
188     \if@tempswa\@tempdima=.5\@tempdima\fi%
189     \edef\SX@width{\the\@tempdima}%
190     \fi
    Correct \SX@width if a frame is requested.
191     \ifx\SX@rframe\@empty
192         \long\def\SX@frame##1{##1}%
193     \else
194         \let\SX@frame\fbbox
195         \setlength\@tempdima{\SX@width-2\fbboxsep-2\fbboxrule}%
196         \edef\SX@width{\the\@tempdima}%
197     \fi
198     \isSX@odd{\def\@tempa{l}}{\def\@tempa{r}}%
199     \makebox[\linewidth][\@tempa]{%
200         \parbox{\linewidth+\SX@overhang}{%
\SX@codefile (\jobname.tmp) is not necessary for the filelist.
201         \let\@addtofilelist\@gobble
202         \let\lst@ifdisplaystyle=\iftrue
203         \SX@KillAboveCaptionskip\lst@MakeCaption{t}%
204         \lst@belowskip=\z@
    Use the “natural” width of the result code if “varwidth” is true. .
205         \let\SX@MakeCaption\lst@MakeCaption
206         \let\lst@MakeCaption\@gobble{}

207         \setbox\SX@ResBox\hbox{%
208             \SX@frame{%
209                 \@nameuse{\if@SX@varwidth varwidth\else minipage\fi}%
210                 \SX@width\relax
211                 \begingroup
212                     \SX@resultInput
213                 \endgroup
214                 \@nameuse{end\if@SX@varwidth varwidth\else minipage\fi}}}%
215         \edef\SX@width{\the\wd\SX@ResBox}%
216         \ifundefined{SX@put@SX@pos}%
217             {\@latex@error{Parameter ‘\SX@pos’ undefined}\@ehd}%
218             {\@nameuse{SX@put@SX@pos}%
219              {\SX@width}{\box\SX@ResBox}{\SX@codeInput}}}%
220         \let\lst@MakeCaption\SX@MakeCaption
221         \lst@MakeCaption{b}\SX@KillBelowCaptionskip
222     }%
223 }%
224 \endtrivlist
225 \ifx\SX@float\relax\else\expandafter\lst@endfloat\fi
226 \gdef\SX@@explpreset{}%
227 \endgroup
228 }

229 \newcommand\SX@SkipToFirst{%

```

```

230 \ifeof\@inputcheck\else
231 \ifnum \lst@lineno=\lst@firstline\else
232 \readline\@inputcheck to\SX@tempa
233 \typeout{IGNORE (\the\lst@lineno)}%
234 \global\advance\lst@lineno\@ne
235 \SX@SkipToFirst
236 \fi
237 \fi
238 }
239 \newcommand\SX@ProcessResult{%
240 \ifeof\@inputcheck
241 \let\SX@tempb\relax
242 \else
243 \let\SX@tempb\SX@ProcessResult
244 \ifnum \lst@lineno>\lst@lastline\relax
245 \ifx\lst@linerange\@empty
246 \let\SX@tempb\relax
247 \else
248 \lst@GetLineInterval
249 \SX@SkipToFirst
250 \fi
251 \else
252 \readline\@inputcheck to\SX@tempa
253 \typeout{READ (\the\lst@lineno)}%
254 \expandafter\g@addto@macro
255 \expandafter\SX@lines\expandafter{\SX@tempa^^J}%
256 \global\advance\lst@lineno\@ne
257 \fi
258 \fi
259 \SX@tempb
260 }

```

\SX@input

```

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

```



```

282 \scantokens\expandafter{\SX@lines}%
283 \endgroup
284 }

285 \newcommand*\SX@Info{%
286 \typeout{-----}%
287 \typeout{pos=\SX@pos}%
288 \typeout{width=\SX@width}%
289 \typeout{hsep=\SX@hsep}%
290 \typeout{vsep=\SX@vsep}%
291 \typeout{overhang=\SX@overhang}%
292 \typeout{rframe=\SX@rframe}%
293 \typeout{codefile=\SX@codefile}%
294 \@ifundefined{lst@firstline}{}%
295 {\typeout{\string\lst@firstline=\lst@firstline}}%
296 \@ifundefined{lst@lastline}{}%
297 {\typeout{\string\lst@lastline=\lst@lastline}}%
298 \@ifundefined{lst@linrange}{}%
299 {\typeout{\string\lst@linrange=\lst@linrange}}%
300 \typeout{\string\if@SX@wide=\if@SX@wide TRUE\else FALSE\fi}%
301 \typeout{\string\if@SX@rangeaccept=\if@SX@rangeaccept TRUE\else FALSE\fi}%
302 \typeout{\string\if@SX@varwidth=\if@SX@varwidth TRUE\else FALSE\fi}%
303 \typeout{graphicfile=\SX@graphicname, graphicparameter=[\SX@graphicparam]}%
304 \typeout{-----}%
305 }
306 \providecommand*\MakePercentIgnore{\catcode'\%9\relax}
307 \providecommand*\MakePercentComment{\catcode'\%14\relax}

```

\SX@resultInput

```

308 \newcommand*\SX@resultInput{%
309 \ifx\SX@graphicname\empty
310 \begingroup
311 \MakePercentComment\makeatother\catcode'\%M=5\relax
312 \SX@@preset\SX@preset
313 \if@SX@rangeaccept
314 \let\SX@tempa=\SX@input
315 \else
316 \let\SX@tempa=\input
317 \fi
318 \SX@tempa{\SX@codefile}\par%
319 \endgroup
320 \else
321 \expandafter\includegraphics\expandafter[\SX@graphicparam]%
322 {\SX@graphicname}%
323 \fi
324 }

```

\SX@codeInput

```

325 \newcommand*\SX@codeInput{%
  Without a caption entry the command \lstinputlisting adds the filename to
  the “list of listings” (lol). This should be avoided.
326 \begingroup
  The default parameters for all examples.
327 \expandafter\lstset\expandafter{\SX@explpreset}%

```

If "numbers=none" then margin dimensions should be zero.

```

328 \expandafter\lstset\expandafter{\SX@@explpreset}%
329 \ifx\lst@PlaceNumber\@empty
330 \g@addto@macro\SX@@explpreset{,xleftmargin=0pt,xrightmargin=0pt}%
331 \fi
332 \SX@Info
333 \expandafter\lstinputlisting\expandafter%
334 [\SX@@explpreset,nolol=true,caption={}]{\SX@codefile}%
335 \endgroup
336 }%

```

```

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

```

All the default values.

```

340 \lstset{explpreset={numbers=left,numberstyle=\tiny,numbersep=.3em,
Negative width means defaults.
341 xleftmargin=1em,columns=flexible,language=[LaTeX]TEX},pos=1,width=-99pt,
342 overhang=0pt,hsep=\columnsep,vsep=\bigskipamount,rframe=single}
.
343 \AtBeginDocument{%
344 \def\theHlstnumber{\thelstlisting.\arabic{lstnumber}.\lst@neglisting}%
345 }

```

Changing the defaults possible in showexpl.cfg.

```

346 \InputIfFileExists{showexpl.cfg}{-}{-}

```

Change History

v0.1a	General: "hpos" and "vpos" added, "pos" removed (RN). 3	New macro \LTXinputExample (RN). 10
	Initial version 1	LTXexample: Renamed from "exam- ple" to "LTXexample" (RN). . . 6
v0.1b	\SX@put@t/b/l/r/o/i: Positioning the captions more independend of the result and code area (RN). 5	v0.1i General: Better caption positioning and correct distance between the parts (RN). 6
v0.1c	\SX@put@t/b/l/r/o/i: Commands \SX@KillAboveCaptionskip and \SX@KillBelowCaptionskip added (RN). 5	v0.1j General: "rangeaccept" added (RN). 3 \SX@input: For ranges of lines (RN). 8
v0.1f	General: "lstpreset" added. (RN). 3	v0.1k General: Some bug corrections (RN). 3
v0.1h	General: "codefile" added. (RN). . 3 "lstpreset" renamed to "explpre- set" (RN). 3	v0.1l \SX@put@t/b/l/r/o/i: Change [a]bove to [t]op (RN). 5 v0.1l General: "graphic" added (RN). . . 3

v0.1m	General: Problem related to \label/\ref solved (RN).	6	v0.3e	\SX@@preset: More redefinitions added (RN).	4
v0.2a	General: “varwidth” and “justifica- tion” added (RN).	3	v0.3g	General: \SX@ProcessResult is now working correctly using \readline and \scantokens. Thanks to Ulrich Diez for help (RN).	7
v0.2b	General: Check if \SX@put@? is de- fined (RN).	6		Missing \newcommand for \SX@@explpreset added (RN).	4
v0.3a	General: “attachfile” added (RN).	3	v0.3h	General: New Option ‘attachfiles’ (RN).	3
	\SX@attachfile: Attach file func- tionality (with pdfTeX) added (RN).	4	v0.3j	\SX@put@code@result: Setting \lst@MakeCaption to was a bad idea for hyperlinks. Group added to varwidth environ- ment. (Suggestions by Ulrike Fischer.).	7
v0.3b	\SX@resultInput: Input of re- sult code now inside a group; \makeatother added (RN).	9	v0.3k	General: Definition for “hyperref” (suggested by Heiko Oberdiek)	10
v0.3c	\SX@resultInput: Wrong catcode for newline char corrected (RN).	9		\SX@put@code@result: Setting \lst@MakeCaption to \gobble again (prevent multiply defined labels; label key)	7
v0.3d	\SX@resultInput: Missing \par added (RN).	9			

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	\@latexerr	267	C	
\%	\@temptokena	145, 154	\c@lstlisting	147
\@input	\^	278, 311	\c@ltxexample	147
\@SX@attachfilefalse			\cite	64
\@SX@rangeacceptfalse	A		\closein	281

\filename@base	268	\lst@linerange	245, 271, 273, 274, 299	\SX@@explpreset 55, 154, 167, 170, 226, 272, 328, 330, 334, 338
\filename@ext	266, 268	\lst@MakeCaption . . .	203, 205, 206, 220, 221	\SX@preset	<u>58</u> , 312
\filename@parse	265	\lst@neglisting	344	\SX@attachfile 11, <u>90</u> , 129
\footnote	74	\lst@PlaceNumber . . .	329	\SX@CodeArea	99, 102, 108, 112, 125
G		\lst@RequireAspects . .	16	\SX@codefile	30, 56, 93, 150, 154, 157, 293, 318, 334
\g@addto@macro 170, 254, 330, 338	\lsthk@PreSet	271	\SX@codeInput	219, <u>325</u>
I		\lstinputlisting	333	\SX@def@WD	18–20
\if@SX@attachfile 34, 38, 91	\lstKV@OptArg	42	\SX@defaultWD	<u>17</u> , 180
\if@SX@rangeaccept 31, 35, 301, 313	\lstKV@SetIf 26, 35, 36, 38	\SX@eat@version 59, 60, 79
\if@SX@varwidth . . .	32, 36, 209, 214, 302	\lstnewenvironment . .	143	\SX@explpreset	29, 54, 148, 165, 327
\if@SX@wide 26, 33, 177, 300	\lstset	148, 165, 167, 175, 272, 327, 328, 340	\SX@float 166, 169, 171, 225
\if@twoside	81	LTxexample (environ- ment)	1, 143	\SX@frame	192, 194, 208
\ifeof	230, 240	\LTXinputExample . . .	1, 337	\SX@graphicname 39, 43, 151, 155, 303, 309, 322
\IfFileExists	10, 263	M		\SX@graphicparam 40, 43, 152, 155, 303, 321
\ifSX@wasodd	80, 115, 118	\makeatother	311	\SX@hsep	23, 49, 107, 111, 289
\ifthenelse	83	\makebox	199	\SX@IDENT	83, 89, 179
\immediate	71	\MakePercentComment 307, 311	\SX@Info	277, 285, 332
\includegraphics . . .	321	\MakePercentIgnore . .	306	\SX@input	<u>261</u> , 314
\index	68	\marginpar	73	\SX@justification 37, 57, 121
\isodd	83	\marginparsep	177	\SX@KillAboveCaptionskip	.. 131, 203
\isSX@odd	<u>80</u> , 198	\marginparwidth	177	\SX@KillBelowCaptionskip	.. 137, 221
L		N		\SX@lines	255, 279, 282
\label	68, 179	\newbox	46	\SX@MakeCaption . . .	205, 220
\listoffigures	65	\newlinechar	278	\SX@overhang	25, 51, 177, 187, 200, 291
\listoftables	66	O		\SX@pos	21, 47, 184, 186, 216–218, 287
\lst@beginfloat	171	\openin	270	\SX@preset	28, 53, 312
\lst@BeginWriteFile . .	157	P		\SX@ProcessResult 239, 243, 280
\lst@belowskip	204	\pagebreak	97, 102	\SX@put@code@result	.. 161, <u>163</u> , 339
\lst@caption 132, 138, 174	\pageref	83	\SX@put@t	96
\lst@captionpos	133, 139	\pagestyle	70	\SX@put@t/b/l/r/o/i	<u>96</u>
\lst@endfloat	225	\printindex	66	\SX@ResBox 46, 207, 215, 219
\lst@EndWriteFile . . .	160	R			
\lst@firstline 231, 274, 295	\raggedright	57		
\lst@float	166, 168, 169	\raisebox	129		
\lst@GetLineInterval 248, 276	\readline	232, 252		
\lst@ifdisplaystyle . . .	202	\rlap	129		
\lst@ifSubstring 133, 139	S			
\lst@Key	21–30, 35–38, 41	\sbox	127		
\lst@lastline 244, 274, 297	\scantokens	282		
\lst@lineno 231, 233, 234, 244, 253, 256, 271	\stepcounter	179		
		\string	295, 297, 299–302		

\SX@ResultArea . 97, 104, 108, 112, 120	. 241, 243, 246, 259	T
\SX@resultInput 212, <u>308</u>	\SX@vsep	\theHlstnumber 344
\SX@rframe	24, 50, 98, 103, 290	\thelstlisting 344
. . 27, 52, 191, 292	\SX@wasoddfalse . . . 84	\theltxexample 92
\SX@SkipToFirst . . .	\SX@wasoddtrue . . 84, 86	\thispagestyle 70
. 229, 235, 249, 279	\SX@width	U
\SX@tempa . 232, 252, 255, 314, 316, 318	22, 48, 180, 181, 189, 195, 196,	\usebox 128
\SX@tempb	210, 215, 219, 288	W
		\write 71