

# The `showexpl` package\*

Rolf Niepraschk (Rolf.Niepraschk@ptb.de)

2012/09/22

## 1 Introduction

The documentation of a  $\text{\LaTeX}$  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  $\text{\LaTeX}$  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` v0.3j, dated 2012/09/22.

- 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<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=...**”.

```

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

```

```

226 \newcommand\SX@SkipToFirst{%
227 \ifeof\@inputcheck\else
228 \ifnum \lst@lineno=\lst@firstline\else
229 \readline\@inputcheck to\SX@tempa
230 \typeout{IGNORE (\the\lst@lineno)}}%

```

```

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

```

\SX@input

```

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

```



```

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

```

\SX@resultInput

```

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

```

\SX@codeInput

```

322 \newcommand*\SX@codeInput{%
  Without a caption entry the command \lstinputlisting adds the filename to
  the “list of listings” (lol). This should be avoided.
323   \begingroup
  The default parameters for all examples.
324   \expandafter\lstset\expandafter{\SX@explpreset}%
  If “numbers=none” then margin dimensions should be zero.
325   \expandafter\lstset\expandafter{\SX@@explpreset}%

```

```

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

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

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

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

```

## Change History

v0.1a	General: “hpos” and “vpos” added, “pos” removed (RN). . . . . 3	v0.1j	General: “rangeaccept” added (RN). . . . . 3
	Initial version . . . . . 1		\SX@input: For ranges of lines (RN). . . . . 8
v0.1b	\SX@put@t/b/l/r/o/i: Positioning the captions more independend of the result and code area (RN). . . . . 5	v0.1k	General: Some bug corrections (RN). . . . . 3
v0.1c	\SX@put@t/b/l/r/o/i: Commands \SX@KillAboveCaptionskip and \SX@KillBelowCaptionskip added (RN). . . . . 5		\SX@put@t/b/l/r/o/i: Change [a]bove to [t]op (RN). . . . . 5
v0.1f	General: “lstpreset” added. (RN). 3	v0.1l	General: “graphic” added (RN). . . 3
v0.1h	General: “codefile” added. (RN). . 3 “lstpreset” renamed to “explpre- set” (RN). . . . . 3 New macro \LTXinputExample (RN). . . . . 10 LTXexample: Renamed from “exam- ple” to “LTXexample” (RN). . . 6	v0.1m	General: Problem related to \label/\ref solved (RN). . . . 6
v0.1i	General: Better caption positioning and correct distance between the parts (RN). . . . . 6	v0.2a	General: “varwidth” and “justifica- tion” added (RN). . . . . 3 “varwidth” package used (RN). . 6
		v0.2b	General: Check if \SX@put@? is de- fined (RN). . . . . 6
		v0.3a	General: “attachfile” added (RN). 3 \SX@attachfile: Attach file func- tionality (with pdfTEX) added (RN). . . . . 4

v0.3b		now working correctly using
\SX@resultInput:	Input of re-	\readline and \scantokens.
	sult code now inside a group;	Thanks to Ulrich Diez for help
	\makeatother added (RN). . . .	(RN). . . . . 7
v0.3c		Missing \newcommand for
\SX@resultInput:	Wrong catcode	\SX@@explpreset added (RN). 4
	for newline char corrected	
	(RN). . . . . 9	v0.3h
v0.3d		General: New Option ‘attachfiles’
\SX@resultInput:	Missing \par	(RN). . . . . 3
	added (RN). . . . . 9	v0.3j
v0.3e		\SX@put@code@result: Setting
\SX@@preset:	More redefinitions	\lst@MakeCaption to \@gobble
	added (RN). . . . . 4	was a bad idea for hyperlinks.
v0.3g		Group added to varwidth envi-
General: \SX@ProcessResult	is	ronment. (Suggestions by UL-
		rike Fischer.). . . . . 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	B	G
\% . . . . . 303, 304	\belowcaptionskip . 140	\g@addto@macro . . . .
\@@input . . . . . 72	\bibliography . . . . 69	. 170, 251, 327, 335
\@SX@attachfilefalse 34	\bigskipamount . . . . 339	
\@SX@rangeacceptfalse	\box . . . . . 217	<b>I</b>
. . . . . 31		\if@SX@attachfile .
\@SX@varwidthfalse . 32	<b>C</b>	. . . . . 34, 38, 91
\@SX@widefalse . . . . 33	\c@lstlisting . . . . 147	\if@SX@rangeaccept .
\@addtofilelist . . . 201	\c@ltxexample . . . . 147	. . . 31, 35, 298, 310
\@capttype . . . . . 62, 63	\cite . . . . . 64	\if@SX@varwidth . . .
\@ehd . . . . . 215, 265	\closein . . . . . 278	32, 36, 207, 212, 299
\@footnotetext . . . . 75	\closeout . . . . . 72	\if@SX@wide . . . . .
\@gobble . . . . .	\columnsep . . . . . 339	. . . 26, 33, 177, 297
. 68–70, 72, 75, 201	<b>E</b>	\if@twoside . . . . . 81
\@gobbletwo . . . . . 71	\endgraf . . . . . 97, 102	\ifeof . . . . . 227, 237
\@inputcheck 227, 229,	environments:	\IfFileExists . . 10, 260
237, 249, 267, 278	LTXexample . . . 1, <u>143</u>	\ifSX@wasodd 80, 115, 118
\@latex@error . . . . 215	<b>F</b>	\ifthenelse . . . . . 83
\@latexerr . . . . . 264	\fbox . . . . . 194	\immediate . . . . . 71
\@temptokena . . 145, 154	\fboxrule . . . . . 195	\includegraphics . . 318
\^ . . . . . 275, 308	\fboxsep . . . . . 195	\index . . . . . 68
<b>A</b>	\filename@area . . . . 265	\isodd . . . . . 83
\abovecaptionskip . 134	\filename@base . . . . 265	\isSX@odd . . . . . <u>80</u> , 198
\abovedisplayskip . . . . . 77	\filename@ext . 263, 265	
\abovedisplayskip . 76	\filename@parse . . . 262	<b>L</b>
\attachfile . . . . . 92	\footnote . . . . . 74	\label . . . . . 68, 179
		\listoffigures . . . . 65
		\listoftables . . . . 66
		\lst@beginfloat . . . 171
		\lst@BeginWriteFile 157

<code>\lst@belowskip</code> . . . . .	204				
<code>\lst@caption</code> . . . . .					
. . . . .	132, 138, 174				
<code>\lst@captionpos</code> 133, 139					
<code>\lst@endfloat</code> . . . . .	222				
<code>\lst@EndWriteFile</code> . .	160				
<code>\lst@firstline</code> . . . . .					
. . . . .	228, 271, 292				
<code>\lst@float</code> 166, 168, 169					
<code>\lst@GetLineInterval</code>					
. . . . .	245, 273				
<code>\lst@ifdisplaystyle</code> 202					
<code>\lst@ifSubstring</code> . .					
. . . . .	133, 139				
<code>\lst@Key</code> 21–30, 35–38, 41					
<code>\lst@lastline</code> . . . . .					
. . . . .	241, 271, 294				
<code>\lst@lineno</code> . . . . .					
. . . . .	228, 230, 231,				
241, 250, 253, 268					
<code>\lst@linerange</code> 242,					
268, 270, 271, 296					
<code>\lst@MakeCaption</code> . .					
. . . . .	203, 218				
<code>\lst@PlaceNumber</code> . .	326				
<code>\lst@RequireAspects</code> 16					
<code>\lsthk@PreSet</code> . . . . .	268				
<code>\lstinputlisting</code> . .	330				
<code>\lstKV@OptArg</code> . . . . .	42				
<code>\lstKV@SetIf</code> . . . . .					
. . . . .	26, 35, 36, 38				
<code>\lstnewenvironment</code> .	143				
<code>\lstset</code> . . . . .	148,				
165, 167, 175,					
269, 324, 325, 337					
L <sup>T</sup> Xexample (environ-					
ment) . . . . .	1, 143				
L <sup>T</sup> XinputExample 1, 334					
<b>M</b>					
<code>\makeatother</code> . . . . .	308				
<code>\makebox</code> . . . . .	199				
<code>\MakePercentComment</code>					
. . . . .	304, 308				
<code>\MakePercentIgnore</code> .	303				
<code>\marginpar</code> . . . . .	73				
<code>\marginparsep</code> . . . . .	177				
<code>\marginparwidth</code> . . .	177				
<b>N</b>					
<code>\newbox</code> . . . . .	46				
<code>\newlinechar</code> . . . . .	275				
<b>O</b>					
<code>\openin</code> . . . . .	267				
<b>P</b>					
<code>\pagebreak</code> . . . . .	97, 102				
<code>\pageref</code> . . . . .	83				
<code>\pagestyle</code> . . . . .	70				
<code>\printindex</code> . . . . .	66				
<b>R</b>					
<code>\raggedright</code> . . . . .	57				
<code>\raisebox</code> . . . . .	129				
<code>\readline</code> . . . . .	229, 249				
<code>\rlap</code> . . . . .	129				
<b>S</b>					
<code>\sbox</code> . . . . .	127				
<code>\scantokens</code> . . . . .	279				
<code>\stepcounter</code> . . . . .	179				
<code>\string</code> 292, 294, 296–299					
<code>\SX@@explpreset</code> . . .					
. . . . .	55, 154, 167,				
170, 223, 269,					
325, 327, 331, 335					
<code>\SX@@preset</code> . . . . .	58, 309				
<code>\SX@attachfile</code> . . . .					
. . . . .	11, 90, 129				
<code>\SX@CodeArea</code> . . . . .	99,				
102, 108, 112, 125					
<code>\SX@codefile</code> . . . . .	30,				
56, 93, 150, 154,					
157, 290, 315, 331					
<code>\SX@codeInput</code> . . . . .	217, 322				
<code>\SX@def@WD</code> . . . . .	18–20				
<code>\SX@defaultWD</code> . . . .	17, 180				
<code>\SX@eat@version</code> . . . .					
. . . . .	59, 60, 79				
<code>\SX@explpreset</code> . . . .					
29, 54, 148, 165, 324					
<code>\SX@float</code> . . . . .					
. . . . .	166, 169, 171, 222				
<code>\SX@frame</code> . . . . .	192, 194, 206				
<code>\SX@graphicname</code> . . . .					
. . . . .	39, 43, 151,				
155, 300, 306, 319					
<code>\SX@graphicparam</code> . . .					
. . . . .	40, 43,				
152, 155, 300, 318					
<code>\SX@hsep</code> . . . . .					
23, 49, 107, 111, 286					
<code>\SX@IDENT</code> . . . . .	83, 89, 179				
<code>\SX@Info</code> . . . . .	274, 282, 329				
<code>\SX@input</code> . . . . .	258, 311				
<code>\SX@justification</code> . .					
. . . . .	37, 57, 121				
<code>\SX@KillAboveCaptionskip</code>					
. . . . .	131, 203				
<code>\SX@KillBelowCaptionskip</code>					
. . . . .	137, 218				
<code>\SX@lines</code> . . . . .	252, 276, 279				
<code>\SX@overhang</code> . . . . .	25, 51,				
177, 187, 200, 288					
<code>\SX@pos</code> . . . . .	21, 47, 184,				
186, 214–216, 284					
<code>\SX@preset</code> . . . . .	28, 53, 309				
<code>\SX@ProcessResult</code> . .					
. . . . .	236, 240, 277				
<code>\SX@put@code@result</code>					
. . . . .	161, 163, 336				
<code>\SX@put@t</code> . . . . .	96				
<code>\SX@put@t/b/l/r/o/i</code> <u>96</u>					
<code>\SX@ResBox</code> . . . . .					
. . . . .	46, 205, 213, 217				
<code>\SX@ResultArea</code> . . . .	97,				
104, 108, 112, 120					
<code>\SX@resultInput</code> 210, <u>305</u>					
<code>\SX@rframe</code> . . . . .					
. . . . .	27, 52, 191, 289				
<code>\SX@SkipToFirst</code> . . . .					
. . . . .	226, 232, 246, 276				
<code>\SX@tempa</code> . . . . .	229, 249,				
252, 311, 313, 315					
<code>\SX@tempb</code> . . . . .					
. . . . .	238, 240, 243, 256				
<code>\SX@vsep</code> . . . . .					
24, 50, 98, 103, 287					
<code>\SX@wasoddfalse</code> . . .	84				
<code>\SX@wasoddtrue</code> . . .	84, 86				
<code>\SX@width</code> . . . . .					
22, 48, 180, 181,					
189, 195, 196,					
208, 213, 217, 285					
<b>T</b>					
<code>\theltxexample</code> . . . .	92				
<code>\thispagestyle</code> . . . .	70				
<b>U</b>					
<code>\usebox</code> . . . . .	128				
<b>W</b>					
<code>\write</code> . . . . .	71				