

The `showexpl` package*

Rolf Niepraschk (Rolf.Niepraschk@gmx.de)

2014/01/19

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[\langle key val list \rangle]{\langle file \rangle}
```

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

```
\begin{LTXexample}[\langle key val list \rangle]\dots\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` v0.3l, dated 2014/01/19.

- 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 `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 *⟨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.
- scaled** Without a value the formatted text will be scaled to fit the given width of the result area. With a number as value the formatted text will be scaled by this number.

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 \newcommand*\SX@scaled{}
30 \lst@Key{scaled}{?}[!]{\def\SX@scaled{#1}}

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

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

38 \lst@Key{varwidth}f[t]{\lstKV@SetIf{#1}\if@SX@varwidth}
39 \lst@Key{justification}\relax{\def\SX@justification{#1}}
40 \lst@Key{attachfile}f[t]{\lstKV@SetIf{#1}\if@SX@attachfile}
41 \newcommand*\SX@graphicname{}%
42 \newcommand*\SX@graphicparam{}%

```

```

43 \lst@Key{graphic}{ } [] {%
44   \lstKV@OptArg[width=\linewidth]{#1}{%
45     \edef\SX@graphicparam{##1}\edef\SX@graphicname{##2}%
46   }%
47 }%
48 \newbox\SX@ResBox
49 \newcommand*\SX@pos{}
50 \newcommand*\SX@width{}
51 \newcommand*\SX@hsep{}
52 \newcommand*\SX@vsep{}
53 \newcommand*\SX@overhang{}
54 \newcommand*\SX@rframe{}
55 \newcommand\SX@preset{}
56 \newcommand*\SX@explpreset{}

57 \newcommand*\SX@@explpreset{}
58 \newcommand*\SX@codefile{}\edef\SX@codefile{\jobname.tmp}
59 \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=...`”.

```

60 \newcommand*\SX@@preset{%
61   \renewcommand\documentclass[2] [] {\SX@eat@version}%
62   \renewcommand\usepackage[2] [] {\SX@eat@version}%
63   \renewenvironment{document}{ } {%
64     \renewenvironment{figure}[1] [] {\def\@cuptype{figure}} {%
65     \renewenvironment{table}[1] [] {\def\@cuptype{table}} {%
66     \renewcommand\cite[1] [] {%
67     \let\tableofcontents\relax \let\listoffigures\relax
68     \let\listoftables\relax \let\printindex\relax
69     \let\listfiles\relax \let\nofiles\relax
70     \let\index@gobble \let\label@gobble
71     \let\bibliography@gobble
72     \let\pagestyle@gobble \let\thispagestyle@gobble
73     %%\let\immediate\relax \let\write@gobbletwo
74     %%\let\closeout@gobble \let\@input@gobble
75     \renewcommand\marginpar[2] [] {%
76     \renewcommand\footnote[2] [] {%
77     \let\@footnotetext@gobble
78     %%\abovedisplayskip=\z@
79     %%\abovedisplayshortskip=\z@
80   }
81 \newcommand*\SX@eat@version[1] [] {}

```

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

```

82 \newif\ifSX@wasodd
83 \if@twoside
84   \newcommand*\isSX@odd}[2] {%
85     \ifthenelse{\isodd{\pageref{\SX@IDENT}}}{%
86       {SX@wasoddtrue #1}{SX@wasoddfalse #2}}
87   \else
88     \newcommand*\isSX@odd}[2] {#1}\SX@wasoddtrue
89 \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.

```
90 \newcounter{ltxexample}
91 \newcommand*{\SX@IDENT}{SX@number\value{ltxexample}}
```

`\SX@attachfile`

```
92 \newcommand*\SX@attachfile{%
93   \ifSX@attachfile
94     \attachfile[mimetype=text/plain,subject={example \theltxexample}]%
95     {\SX@codefile}{}%
96   \fi
97 }
```

`\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.

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

```

135 \lst@ifsubstring t\lst@captionpos
136   {\vskip-\abovecaptionskip}{}%
137 \fi
138 }
139 \newcommand*\SX@KillBelowCaptionskip{%
140   \ifx\lst@caption\@empty\else
141     \lst@ifsubstring b\lst@captionpos
142       {\vskip-\belowcaptionskip}{}%
143     \fi
144 }

```

LTXexample

```

145 \lstnewenvironment{LTXexample}[1] []
146 {%
147   \@temptokena{#1}%
148   \beginingroup

```

For "codefile=..."/"graphic=..." if \theltxexample or \thelstlisting is part of the filename.

```

149   \advance\c@ltxexample\@ne \advance\c@lstlisting\@ne
150   \expandafter\lstset\expandafter{\SX@explpreset,#1}%
151   \edef\x{\endgroup
152     \def\noexpand\SX@codefile{\SX@codefile}%
153     \def\noexpand\SX@graphicname{\SX@graphicname}%
154     \def\noexpand\SX@graphicparam{\SX@graphicparam}}%
155   \x
156   \xdef\SX@@explpreset{\the\@temptokena,codefile=\SX@codefile,
157     graphic={[\SX@graphicparam]{\SX@graphicname}}}%
158   \setbox\@tempboxa=\hbox\bgroup% Warum noetig?
159   \lst@BeginWriteFile{\SX@codefile}%
160 }
161 {%
162   \lst@EndWriteFile\egroup
163   \SX@put@code@result
164 }

```

\SX@put@code@result

```

165 \newcommand*\SX@put@code@result{%
166   \beginingroup
167   \expandafter\lstset\expandafter{\SX@explpreset}%
168   \let\lst@float=\relax\let\SX@float=\relax

```

Without the following call \lst@beginfloat is undefined.

```

169   \expandafter\lstset\expandafter{\SX@@explpreset}%
170   \ifx\lst@float\relax\else

```

\lst@float must be \relax because the whole "example" should float but not the listings part in addition.

```

171     \let\SX@float=\lst@float\let\lst@float=\relax
172     \g@addto@macro\SX@@explpreset{,float=false}%
173     \edef\@tempa{\noexpand\lst@beginfloat{lstlisting}[\SX@float]}%
174     \expandafter\@tempa
175   \fi
176   \ifx\lst@caption\@empty

```

```

177     \lstset{ nolol=true }%
178     \fi
179     \if@SX@wide\def\SX@overhang{\marginparwidth+\marginparsep}\fi
180     \trivlist\item\relax
181     \stepcounter{ ltxexample }\label{\SX@IDENT}%
    Make \SX@width a real dimension if the unit is missing.
182     \SX@defaultWD\SX@width{\SX@width}%
    Set the default width if necessary.
183     \ifdim\SX@width<\z@
184     \@tempwattrue
185     \def\@tempa{t}%
186     \ifx\@tempa\SX@pos\@tempswafalse\fi
187     \def\@tempa{b}%
188     \ifx\@tempa\SX@pos\@tempswafalse\fi
189     \setlength\@tempdima{\linewidth+\SX@overhang}%
190     \if@tempswa\@tempdima=.5\@tempdima\fi%
191     \edef\SX@width{\the\@tempdima}%
192     \fi
    Correct \SX@width if a frame is requested.
193     \ifx\SX@rframe\@empty
194     \long\def\SX@frame##1{##1}%
195     \else
196     \let\SX@frame\fbbox
197     \setlength\@tempdima{\SX@width-2\fbboxsep-2\fbboxrule}%
198     \edef\SX@width{\the\@tempdima}%
199     \fi
200     \isSX@odd{\def\@tempa{l}}{\def\@tempa{r}}%
201     \makebox[\linewidth][\@tempa]{%
202     \parbox{\linewidth+\SX@overhang}{%
\SX@codefile (\jobname.tmp) is not necessary for the filelist.
203     \let\@addtofilelist\@gobble
204     \let\lst@ifdisplaystyle=\iftrue
205     \SX@KillAboveCaptionskip\lst@MakeCaption{t}%
206     \lst@belowskip=\z@
    Use the “natural” width of the result code if “varwidth” is true. .
207     \let\SX@MakeCaption\lst@MakeCaption
208     \let\lst@MakeCaption\@gobble{
209     \setbox\SX@ResBox\hbox{%
210     \SX@frame{%
211     \@nameuse{\if@SX@varwidth varwidth\else minipage\fi}%
212     \SX@width\relax
213     \begingroup
214     \SX@resultInput
215     \endgroup
216     \@nameuse{end\if@SX@varwidth varwidth\else minipage\fi}}}%
217     \edef\SX@width{\the\wd\SX@ResBox}%
218     \@ifundefined{SX@put@\SX@pos}%
219     {\@latex@error{Parameter ‘\SX@pos’ undefined}\@ehd}%
220     {\@nameuse{SX@put@\SX@pos}%
221     {\SX@width}{\box\SX@ResBox}{\SX@codeInput}}%

```

```

222         \let\lst@MakeCaption\SX@MakeCaption
223         \lst@MakeCaption{b}\SX@KillBelowCaptionskip
224     }%
225 }%
226 \endtrivlist
227 \ifx\SX@float\relax\else\expandafter\lst@endfloat\fi
228 \gdef\SX@@explpreset{}%
229 \endgroup
230 }

```

```

231 \newcommand\SX@SkipToFirst{%
232 \ifeof\@inputcheck\else
233 \ifnum \lst@lineno=\lst@firstline\else
234 \readline\@inputcheck to\SX@tempa
235 \typeout{IGNORE (\the\lst@lineno)}%
236 \global\advance\lst@lineno\@ne
237 \SX@SkipToFirst
238 \fi
239 \fi
240 }
241 \newcommand\SX@ProcessResult{%
242 \ifeof\@inputcheck
243 \let\SX@tempb\relax
244 \else
245 \let\SX@tempb\SX@ProcessResult
246 \ifnum \lst@lineno>\lst@lastline\relax
247 \ifx\lst@linrange\@empty
248 \let\SX@tempb\relax
249 \else
250 \lst@GetLineInterval
251 \SX@SkipToFirst
252 \fi
253 \else
254 \readline\@inputcheck to\SX@tempa
255 \typeout{READ (\the\lst@lineno)}%
256 \expandafter\g@addto@macro
257 \expandafter\SX@lines\expandafter{\SX@tempa^^J}%
258 \global\advance\lst@lineno\@ne
259 \fi
260 \fi
261 \SX@tempb
262 }

```

\SX@input

```

263 \newcommand\SX@input[1]{%
264 \beginingroup
265 \IfFileExists{#1}{-%
266 {-%
267 \filename@parse{#1}%
268 \ifx\filename@ext\relax \def\filename@ext{tex}\fi
269 \@latexerr{File
270 ‘\filename@area\filename@base.\filename@ext’ not found.^^J^^J}\@ehd%
271 }%
272 \openin\@inputcheck#1

```



```

273 \lsthk@PreSet\let\lst@linerange\@empty\global\lst@lineno\@ne
274 \expandafter\lstset\expandafter{\SX@@explpreset}%
275 \ifx\lst@linerange\@empty
276   \edef\lst@linerange{\lst@firstline}-{\lst@lastline},}%
277 \fi
278 \lst@GetLineInterval
279 \SX@Info
280 \newlinechar='^^J\relax
281 \SX@SkipToFirst\let\SX@lines\@empty
282 \SX@ProcessResult
283 \closein\@inputcheck
284 \scantokens\expandafter{\SX@lines}%
285 \endgroup
286 }

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

```

\SX@resultInput

```

310 \newcommand*\SX@resultInput{%
311   \ifx\SX@graphicname\@empty
312     \begingroup
313       \MakePercentComment\makeatother\catcode'\^^M=5\relax
314       \SX@@preset\SX@preset
315       \if@SX@rangeaccept
316         \let\SX@tempa=\SX@input
317       \else
318         \let\SX@tempa=\input
319       \fi
320       \if\SX@scaled ?%
321         \let\SX@tempb=\@firstofone
322       \else
323         \if\SX@scaled !%

```

```

324     \def\SX@tempb##1{\resizebox{\SX@width}{!}{##1}}%
325     \else
326     \def\SX@tempb##1{\scalebox{\SX@scaled}{##1}}%
327     \fi
328     \fi
329     \SX@tempb{\SX@tempa{\SX@codefile}}\par
330 \endgroup
331 \else
332 \expandafter\includegraphics\expandafter[\SX@graphicparam]%
333   {\SX@graphicname}%
334 \fi
335 }

```

\SX@codeInput

```

336 \newcommand*\SX@codeInput{%
  Without a caption entry the command \lstinputlisting adds the filename to
  the “list of listings” (lol). This should be avoided.
337 \begingroup
  The default parameters for all examples.
338 \expandafter\lstset\expandafter{\SX@explpreset}%
  If ”numbers=none” then margin dimensions should be zero.
339 \expandafter\lstset\expandafter{\SX@@explpreset}%
340 \ifx\lst@PlaceNumber\@empty
341 \g@addto@macro\SX@@explpreset{,xleftmargin=0pt,xrightmargin=0pt}%
342 \fi
343 \SX@Info
344 \expandafter\lstinputlisting\expandafter%
345   [\SX@@explpreset,nolol=true,caption={}]{\SX@codefile}%
346 \endgroup
347 }%

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

  All the default values.
351 \lstset{explpreset={numbers=left,numberstyle=\tiny,numbersep=.3em,
  Negative width means defaults.
352 xleftmargin=1em,columns=flexible,language=[LaTeX]TEX},pos=1,width=-99pt,
353 overhang=0pt,hsep=\columnsep,vsep=\bigskipamount,rframe=single}
.
354 \AtBeginDocument{%
355 \def\theHlstnumber{\thelstlisting.\arabic{lstnumber}.\lst@neglisting}%
356 }

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

```

Change History

v0.1a	General: “hpos” and “vpos” added, “pos” removed (RN).	3	v0.3a	General: “attachfile” added (RN).	3
	Initial version	1		<code>\SX@attachfile</code> : Attach file func- tionality (with pdf \TeX) added (RN).	5
v0.1b	<code>\SX@put@t/b/l/r/o/i</code> : Positioning the captions more independend of the result and code area (RN).	5	v0.3b	<code>\SX@resultInput</code> : Input of re- sult code now inside a group; <code>\makeatother</code> added (RN).	9
v0.1c	<code>\SX@put@t/b/l/r/o/i</code> : Commands <code>\SX@KillAboveCaptionskip</code> and <code>\SX@KillBelowCaptionskip</code> added (RN).	5	v0.3c	<code>\SX@resultInput</code> : Wrong catcode for newline char corrected (RN).	9
v0.1f	General: “lstpreset” added. (RN).	3	v0.3d	<code>\SX@resultInput</code> : Missing <code>\par</code> added (RN).	9
v0.1h	General: “codefile” added. (RN).	3	v0.3e	<code>\SX@@preset</code> : More redefinitions added (RN).	4
	“lstpreset” renamed to “explpre- set” (RN).	3	v0.3g	General: <code>\SX@ProcessResult</code> is now working correctly using <code>\readline</code> and <code>\scantokens</code> . Thanks to Ulrich Diez for help (RN).	8
	New macro <code>\LTXinputExample</code> (RN).	10		Missing <code>\newcommand</code> for <code>\SX@@explpreset</code> added (RN).	4
	<code>LTXexample</code> : Renamed from “exam- ple” to “LTXexample” (RN).	6	v0.3h	General: New Option ‘attachfiles’ (RN).	3
v0.1i	General: Better caption positioning and correct distance between the parts (RN).	6	v0.3j	<code>\SX@put@code@result</code> : Setting <code>\lst@MakeCaption</code> to was a bad idea for hyperlinks. Group added to <code>varwidth</code> environ- ment. (Suggestions by Ulrike Fischer.).	7
v0.1j	General: “rangeaccept” added (RN).	3	v0.3k	General: Definition for “hyperref” (suggested by Heiko Oberdiek) 10	
	<code>\SX@input</code> : For ranges of lines (RN).	8		<code>\SX@put@code@result</code> : Setting <code>\lst@MakeCaption</code> to <code>@gobble</code> again (prevent multiply defined labels; label key)	7
v0.1k	General: Some bug corrections (RN).	3	v0.3l	General: Option “scaled” and <code>\SX@scaled</code> added (RN).	3
	<code>\SX@put@t/b/l/r/o/i</code> : Change [a]bove to [t]op (RN).	5		<code>\SX@resultInput</code> : Code for “scaled” option (RN).	9
v0.1l	General: “graphic” added (RN).	3			
v0.1m	General: Problem related to <code>\label/\ref</code> solved (RN).	6			
v0.2a	General: “varwidth” and “justifica- tion” added (RN).	3			
	“varwidth” package used (RN).	6			
v0.2b	General: Check if <code>\SX@put@?</code> is de- fined (RN).	6			

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	environments:	<code>\lst@EndWriteFile</code> . 162
<code>\%</code> 308, 309	<code>LTXexample</code> ... <i>1</i> , <u>145</u>	<code>\lst@firstline</code>
<code>\@input</code> 74	F 233, 276, 297
<code>\@SX@attachfilefalse</code> 36	<code>\fbox</code> 196	<code>\lst@float</code> 168, 170, 171
<code>\@SX@rangeacceptfalse</code>	<code>\fboxrule</code> 197	<code>\lst@GetLineInterval</code>
..... 33	<code>\fboxsep</code> 197 250, 278
<code>\@SX@varwidthfalse</code> . 34	<code>\filename@area</code> 270	<code>\lst@ifdisplaystyle</code> 204
<code>\@SX@widefalse</code> 35	<code>\filename@base</code> 270	<code>\lst@ifsubstring</code> ..
<code>\@addtofilelist</code> ... 203	<code>\filename@ext</code> . 268, 270 135, 141
<code>\@capttype</code> 64, 65	<code>\filename@parse</code> ... 267	<code>\lst@Key</code> 21–28,
<code>\@ehd</code> 219, 270	<code>\footnote</code> 76	30–32, 37–40, 43
<code>\@firstofone</code> 321	G	<code>\lst@lastline</code>
<code>\@footnotetext</code> 77	<code>\g@addto@macro</code> 246, 276, 299
<code>\@gobble</code> 70–	. 172, 256, 341, 349	<code>\lst@lineno</code>
72, 74, 77, 203, 208	I	. 233, 235, 236,
<code>\@gobbletwo</code> 73	<code>\if@SX@attachfile</code> .	246, 255, 258, 273
<code>\@inputcheck</code> 232, 234, 36, 40, 93	<code>\lst@linerange</code> 247,
242, 254, 272, 283	<code>\if@SX@rangeaccept</code> .	273, 275, 276, 301
<code>\@latex@error</code> 219	.. 33, 37, 303, 315	<code>\lst@MakeCaption</code> 205,
<code>\@latexerr</code> 269	<code>\if@SX@varwidth</code> ...	207, 208, 222, 223
<code>\@temptokena</code> .. 147, 156	34, 38, 211, 216, 304	<code>\lst@neglisting</code> ... 355
<code>\^</code> 280, 313	<code>\if@SX@wide</code>	<code>\lst@PlaceNumber</code> .. 340
A	.. 26, 35, 179, 302	<code>\lst@RequireAspects</code> 16
<code>\abovecaptionskip</code> . 136	<code>\if@twoside</code> 83	<code>\lsthk@PreSet</code> 273
<code>\abovedisplayshortskip</code>	<code>\ifeof</code> 232, 242	<code>\lstinputlisting</code> .. 344
..... 79	<code>\ifFileExists</code> .. 10, 265	<code>\lstKV@OptArg</code> 44
<code>\abovedisplayskip</code> . 78	<code>\ifSX@wasodd</code> 82, 117, 120	<code>\lstKV@SetIf</code>
<code>\arabic</code> 355	<code>\ifthenelse</code> 85 26, 37, 38, 40
<code>\attachfile</code> 94	<code>\immediate</code> 73	<code>\lstnewenvironment</code> . 145
B	<code>\includegraphics</code> .. 332	<code>\lstset</code> 150,
<code>\belowcaptionskip</code> . 142	<code>\index</code> 70	167, 169, 177,
<code>\bibliography</code> 71	<code>\isodd</code> 85	274, 338, 339, 351
<code>\bigskipamount</code> 353	<code>\isSX@odd</code> <u>82</u> , 200	<code>LTXexample</code> (environ-
<code>\box</code> 221	L	ment) <i>1</i> , <u>145</u>
C	<code>\label</code> 70, 181	<code>LTXinputExample</code> <i>1</i> , 348
<code>\c@lstlisting</code> 149	<code>\listoffigures</code> 67	M
<code>\c@ltxexample</code> 149	<code>\listoftables</code> 68	<code>\makeatother</code> 313
<code>\cite</code> 66	<code>\lst@beginfloat</code> ... 173	<code>\makebox</code> 201
<code>\closein</code> 283	<code>\lst@BeginWriteFile</code> 159	<code>\MakePercentComment</code>
<code>\closeout</code> 74	<code>\lst@belowskip</code> 206 309, 313
<code>\columnsep</code> 353	<code>\lst@caption</code>	<code>\MakePercentIgnore</code> . 308
E 134, 140, 176	<code>\marginpar</code> 75
<code>\endgraf</code> 99, 104	<code>\lst@captionpos</code> 135, 141	<code>\marginparsep</code> 179
	<code>\lst@endfloat</code> 227	<code>\marginparwidth</code> ... 179
		N
		<code>\newbox</code> 48

<code>\newlinechar</code>	280	<code>\SX@defaultWD</code> . .	<u>17</u> , 182	<code>\SX@put@t/b/l/r/o/i</code>	<u>98</u>
O					
<code>\openin</code>	272	<code>\SX@eat@version</code>	61, 62, 81	<code>\SX@ResBox</code>	48, 209, 217, 221
P					
<code>\pagebreak</code>	99, 104	<code>\SX@explpreset</code>	31, 56, 150, 167, 338	<code>\SX@ResultArea</code> .	99, 106, 110, 114, 122
<code>\pageref</code>	85	<code>\SX@float</code>	168, 171, 173, 227	<code>\SX@resultInput</code>	214, <u>310</u>
<code>\pagestyle</code>	72	<code>\SX@frame</code> .	194, 196, 210	<code>\SX@rframe</code>	27, 54, 193, 294
<code>\printindex</code>	68	<code>\SX@graphicname</code>	41, 45, 153, 157, 305, 311, 333	<code>\SX@scaled</code>	29, 30, 320, 323, 326
R					
<code>\raggedright</code>	59	<code>\SX@graphicparam</code>	42, 45, 154, 157, 305, 332	<code>\SX@SkipToFirst</code>	231, 237, 251, 281
<code>\raisebox</code>	131	<code>\SX@hsep</code>	23, 51, 109, 113, 291	<code>\SX@tempa</code> .	234, 254, 257, 316, 318, 329
<code>\readline</code>	234, 254	<code>\SX@IDENT</code> . .	85, 91, 181	<code>\SX@tempb</code>	243, 245, 248, 261, 321, 324, 326, 329
<code>\resizebox</code>	324	<code>\SX@Info</code> . .	279, 287, 343	<code>\SX@vsep</code>	24, 52, 100, 105, 292
<code>\rlap</code>	131	<code>\SX@input</code>	<u>263</u> , 316	<code>\SX@wasoddfalse</code>	86
S					
<code>\sbox</code>	129	<code>\SX@justification</code>	39, 59, 123	<code>\SX@wasoddttrue</code>	86, 88
<code>\scalebox</code>	326	<code>\SX@KillAboveCaptionskip</code>	133, 205	<code>\SX@width</code>	22, 50, 182, 183, 191, 197, 198, 212, 217, 221, 290, 324
<code>\scantokens</code>	284	<code>\SX@KillBelowCaptionskip</code>	139, 223	T	
<code>\stepcounter</code>	181	<code>\SX@lines</code> .	257, 281, 284	<code>\theHlstnumber</code>	355
<code>\string</code> 297, 299, 301–304		<code>\SX@MakeCaption</code>	207, 222	<code>\thelstlisting</code>	355
<code>\SX@@explpreset</code>	57, 156, 169, 172, 228, 274, 339, 341, 345, 349	<code>\SX@overhang</code> .	25, 53, 179, 189, 202, 293	<code>\theltxexample</code>	94
<code>\SX@@preset</code>	<u>60</u> , 314	<code>\SX@pos</code> .	21, 49, 186, 188, 218–220, 289	<code>\thispagestyle</code>	72
<code>\SX@attachfile</code>	11, <u>92</u> , 131	<code>\SX@preset</code> .	28, 55, 314	U	
<code>\SX@CodeArea</code>	101, 104, 110, 114, 127	<code>\SX@ProcessResult</code>	241, 245, 282	<code>\usebox</code>	130
<code>\SX@codefile</code>	32, 58, 95, 152, 156, 159, 295, 329, 345	<code>\SX@put@code@result</code>	163, <u>165</u> , 350	W	
<code>\SX@codeInput</code> .	221, <u>336</u>	<code>\SX@put@t</code>	98	<code>\write</code>	73
<code>\SX@def@WD</code>	18–20				