

The `showexpl` package*

Rolf Niepraschk (Rolf.Niepraschk@gmx.de)

2020/05/06

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` v0.3q, dated 2020/05/06.

- 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 0pt.
- 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`.
- vsep** 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.

In addition to these options the kind of the result box (default: `\fbox`) can be changed. For example:

```
\renewcommand\ResultBox{\fcolorbox{green}{lightgray}}
\setlength\ResultBoxSep{5mm}% default: \fboxsep
\setlength\ResultBoxRule{2mm}% default: \fboxrule
```

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{refcount,listings,graphicx,varwidth,float}

```

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\ResultBox{} \let\ResultBox=\fbox
50 \newdimen\ResultBoxSep \ResultBoxSep=\fboxsep
51 \newdimen\ResultBoxRule \ResultBoxRule=\fboxrule
52 \newcommand*\SX@pos{}
53 \newcommand*\SX@width{}
54 \newcommand*\SX@hsep{}
55 \newcommand*\SX@vsep{}
56 \newcommand*\SX@overhang{}
57 \newcommand*\SX@rframe{}
58 \newcommand\SX@preset{}
59 \newcommand*\SX@explpreset{}

60 \newcommand*\SX@@explpreset{}
61 \newcommand*\SX@codefile{}\edef\SX@codefile{\jobname.tmp}
62 \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=...**”.

```

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

```

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

```

83 \newif\ifSX@wasodd
84 \if@twoside
85   \newcommand*\isSX@odd{%
86     \begingroup
87       \ifodd\getpagerefnumber{\SX@IDENT}%
88       \aftergroup\SX@wasoddttrue
89     \else

```

```

90     \aftergroup\SX@wasoddfalse
91     \fi
92 \endgroup
93 \ifSX@wasodd
94     \expandafter\@firstoftwo
95 \else
96     \expandafter\@secondoftwo
97 \fi
98 }
99 \else
100 \SX@wasoddtrue
101 \newcommand*\isSX@odd[2]{#1}
102 \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.

```

103 \newcounter{ltxexample}
104 \newcommand*\{SX@IDENT\}{SX@number\value{ltxexample}}

```

`\SX@attachfile`

```

105 \newcommand*\SX@attachfile{%
106     \if@SX@attachfile
107         \attachfile[mimetype=text/plain,subject={example \theltxexample}]{%
108             \{SX@codefile\}}}%
109 \fi
110 }

```

`\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 innner side. Parameter #1 is the width of the result.

```

111 \newcommand*\SX@put@t[3]{%
112     \SX@ResultArea{\linewidth}{#2}\endgraf\pagebreak[2]%
113     \@tempdima=\dimexpr\SX@vsep\vskip\@tempdima
114     \SX@CodeArea{\linewidth}{#3}%
115 }
116 \newcommand*\SX@put@b[3]{%
117     \SX@CodeArea{\linewidth}{#3}\endgraf\pagebreak[2]%
118     \@tempdima=\dimexpr\SX@vsep\vskip\@tempdima
119     \SX@ResultArea{\linewidth}{#2}%
120 }
121 \newcommand*\SX@put@l[3]{%
122     \@tempdimc=\dimexpr\linewidth-#1-\SX@hsep %
123     \SX@ResultArea{#1}{#2}\hfill\SX@CodeArea{\@tempdimc}{#3}%
124 }
125 \newcommand*\SX@put@r[3]{%
126     \@tempdimc=\dimexpr\linewidth-#1-\SX@hsep %
127     \SX@CodeArea{\@tempdimc}{#3}\hfill\SX@ResultArea{#1}{#2}%
128 }
129 \newcommand*\SX@put@o[3]{%
130     \@nameuse{SX@put@\ifSX@wasodd r\else l\fi}{#1}{#2}{#3}%
131 }
132 \newcommand*\SX@put@i[3]{%
133     \@nameuse{SX@put@\ifSX@wasodd l\else r\fi}{#1}{#2}{#3}%
134 }

```

```

135 \newcommand\SX@ResultArea[2]{%
136   \SX@justification\@tempdima=\dimexpr #1 %
137   \parbox\@tempdima{#2}%
138 }
139 \newcommand\SX@CodeArea[2]{%
140   \@tempdima=\dimexpr #1 %
141   \sbox\@tempboxa{\parbox\@tempdima{#2}}%
142   \@tempdima=\dp\@tempboxa\usebox\@tempboxa
143   \rlap{\raisebox{-\@tempdima}[Opt][Opt]{\SX@attachfile}}%
144 }
145 \newcommand*\SX@KillAboveCaptionskip{%
146   \ifx\lst@caption\@empty\else
147     \lst@ifsubstring t\lst@captionpos
148       {\vskip-\abovecaptionskip}{}%
149   \fi
150 }
151 \newcommand*\SX@KillBelowCaptionskip{%
152   \ifx\lst@caption\@empty\else
153     \lst@ifsubstring b\lst@captionpos
154       {\vskip-\belowcaptionskip}{}%
155   \fi
156 }

```

LTXexample

```

157 \lstnewenvironment{LTXexample}[1][ ]{%
158   \@temptokena{#1}%
159   \begingroup

```

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

```

160   \advance\c@ltxexample\@ne \advance\c@lstlisting\@ne
161   \expandafter\lstset\expandafter{\SX@explpreset,#1}%
162   \edef\x{\endgroup
163     \def\noexpand\SX@codefile{\SX@codefile}%
164     \def\noexpand\SX@graphicname{\SX@graphicname}%
165     \def\noexpand\SX@graphicparam{\SX@graphicparam}}%
166   \x
167   \xdef\SX@@explpreset{\the\@temptokena,codefile=\SX@codefile,%
168     graphic={[\SX@graphicparam]{\SX@graphicname}}}%
169   \setbox\@tempboxa=\hbox\bgroup
170   \lst@BeginWriteFile{\SX@codefile}%
171 }
172 {%
173   \lst@EndWriteFile\egroup
174   \SX@put@code@result
175 }

```

\SX@put@code@result

```

176 \newcommand*\SX@put@code@result{%
177   \begingroup
178     \expandafter\lstset\expandafter{\SX@explpreset}%
179     \expandafter\lstset\expandafter{\SX@@explpreset}%

```

Use listings floating procedure if necessary.

```

180 \ifx\lst@float\relax\else
181 \edef\@tempa{\noexpand\lst@beginfloat{lstlisting}[\lst@float]}
182 \expandafter\@tempa
183 \fi
184 \ifx\lst@caption\empty
185 \lstset{lolol=true}%
186 \fi
187 \if@SX@wide\def\SX@overhang{\marginparwidth+\marginparsep}\fi
188 \trivlist\item\relax
189 \stepcounter{ltexample}\label{SX@IDENT}%

```

Make \SX@width a real dimension if the unit is missing.

```

190 \SX@defaultWD\SX@width{\SX@width}%

```

Set the default width if necessary.

```

191 \ifdim\SX@width<\z@
192 \@tempswtrue
193 \def\@tempa{t}%
194 \ifx\@tempa\SX@pos\@tempswfalse\fi
195 \def\@tempa{b}%
196 \ifx\@tempa\SX@pos\@tempswfalse\fi
197 \@tempdima=\dimexpr\linewidth+\SX@overhang %
198 \if@tempswa\@tempdima=.5\@tempdima\fi%
199 \edef\SX@width{\the\@tempdima}%
200 \fi

```

Correct \SX@width if a frame is requested.

```

201 \ifx\SX@rframe\empty
202 \long\def\SX@frame##1{##1}%
203 \else
204 \let\SX@frame\ResultBox
205 \@tempdima=\dimexpr\SX@width-2\ResultBoxSep-2\ResultBoxRule %
206 \edef\SX@width{\the\@tempdima}%
207 \fi
208 \isSX@odd{\def\@tempa{l}}{\def\@tempa{r}}%
209 \makebox[\linewidth][\@tempa]{%
210 \parbox{\dimexpr\linewidth+\SX@overhang}{%

```

\SX@codefile (\jobname.tmp) is not necessary for the filelist.

```

211 \let\@addtofilelist\@gobble
212 \let\lst@ifdisplaystyle=\iftrue
213 \SX@KillAboveCaptionskip\lst@MakeCaption{t}%

```

Use the “natural” width of the result code if “varwidth” is true. .

```

214 \setbox\SX@ResBox\hbox{%
215 \fboxsep\ResultBoxSep
216 \fboxrule\ResultBoxRule
217 \SX@frame{%
218 \@nameuse{\if@SX@varwidth varwidth\else minipage}\fi}%
219 \SX@width\relax
220 \begingroup
221 \SX@resultInput
222 \endgroup
223 \@nameuse{end\if@SX@varwidth varwidth\else minipage}\fi}}%
224 \edef\SX@width{\the\wd\SX@ResBox}%

```

```

225         \@ifundefined{SX@put@SX@pos}%
226         {\@latex@error{Parameter ‘SX@pos’ undefined}\@ehd}%
227         {\@nameuse{SX@put@SX@pos}%
228         {\SX@width}{\box\SX@ResBox}{\SX@codeInput}}}%
229         \lst@MakeCaption{b}\SX@KillBelowCaptionskip
230     }%
231 }%
232 \endtrivlist
233 \ifx\lst@float\relax\else\expandafter\lst@endfloat\fi
234 \gdef\SX@@explpreset{}%
235 \endgroup
236 }

```

```

237 \newcommand\SX@SkipToFirst{%
238 \ifeof\@inputcheck\else
239 \ifnum \lst@lineno=\lst@firstline\else
240 \readline\@inputcheck to\SX@tempa
241 \typeout{IGNORE (\the\lst@lineno)}%
242 \global\advance\lst@lineno\@ne
243 \SX@SkipToFirst
244 \fi
245 \fi
246 }

```

```

247 \newcommand\SX@ProcessResult{%
248 \ifeof\@inputcheck
249 \let\SX@tempb\relax
250 \else
251 \let\SX@tempb\SX@ProcessResult
252 \ifnum \lst@lineno>\lst@lastline\relax
253 \ifx\lst@linerange\@empty
254 \let\SX@tempb\relax
255 \else
256 \lst@GetLineInterval
257 \SX@SkipToFirst
258 \fi
259 \else
260 \readline\@inputcheck to\SX@tempa
261 \typeout{READ (\the\lst@lineno)}%
262 \expandafter\g@addto@macro
263 \expandafter\SX@lines\expandafter{\SX@tempa^^J}%
264 \global\advance\lst@lineno\@ne
265 \fi
266 \fi
267 \SX@tempb
268 }

```

\SX@input

```

269 \newcommand\SX@input[1]{%
270 \begingroup
271 \IfFileExists{#1}{}%
272 {%
273 \filename@parse{#1}%
274 \ifx\filename@ext\relax \def\filename@ext{tex}\fi
275 \@latexerr{File

```



```

276         '\filename@area\filename@base.\filename@ext' not found.^^J^^J}\@ehd%
277     }%
278     \openin\@inputcheck#1
279     \lsthk@PreSet\let\lst@linrange\@empty\global\lst@lineno\@ne
280     \expandafter\lstset\expandafter{\SX@@explpreset}%
281     \ifx\lst@linrange\@empty
282         \edef\lst@linrange{\lst@firstline}-{\lst@lastline},}%
283     \fi
284     \lst@GetLineInterval
285     \SX@Info
286     \newlinechar='^^J\relax
287     \SX@SkipToFirst\let\SX@lines\@empty
288     \SX@ProcessResult
289     \closein\@inputcheck
290     \scantokens\expandafter{\SX@lines}%
291 \endgroup
292 }

293 \newcommand*\SX@Info{%
294     \typeout{-----}%
295     \typeout{pos=\SX@pos}%
296     \typeout{width=\SX@width}%
297     \typeout{hsep=\SX@hsep}%
298     \typeout{vsep=\SX@vsep}%
299     \typeout{overhang=\SX@overhang}%
300     \typeout{rframe=\SX@rframe}%
301     \typeout{codefile=\SX@codefile}%
302     \@ifundefined{lst@firstline}{}%
303     {\typeout{\string\lst@firstline=\lst@firstline}}%
304     \@ifundefined{lst@lastline}{}%
305     {\typeout{\string\lst@lastline=\lst@lastline}}%
306     \@ifundefined{lst@linrange}{}%
307     {\typeout{\string\lst@linrange=\lst@linrange}}%
308     \typeout{\string\if@SX@wide=\if@SX@wide TRUE\else FALSE\fi}%
309     \typeout{\string\if@SX@rangeaccept=\if@SX@rangeaccept TRUE\else FALSE\fi}%
310     \typeout{\string\if@SX@varwidth=\if@SX@varwidth TRUE\else FALSE\fi}%
311     \typeout{graphicfile=\SX@graphicname, graphicparameter=[\SX@graphicparam]}%
312     \typeout{-----}%
313 }

314 \providecommand*\MakePercentIgnore{\catcode'\%9\relax}
315 \providecommand*\MakePercentComment{\catcode'\%14\relax}

```

\SX@resultInput

```

316 \newcommand*\SX@resultInput{%
317     \ifx\SX@graphicname\@empty
318         \begingroup
319             \MakePercentComment\makeatother\catcode'\%M=5\relax
320             \SX@@preset\SX@preset
321             \if@SX@rangeaccept
322                 \let\SX@tempa=\SX@input
323             \else
324                 \let\SX@tempa=\input
325             \fi
326             \if\SX@scaled ?%

```

```

327     \let\SX@tempb=\@firstofone
328   \else
329     \if\SX@scaled !%
330       \def\SX@tempb##1{\resizebox{\SX@width}{!}{##1}}%
331     \else
332       \def\SX@tempb##1{\scalebox{\SX@scaled}{##1}}%
333     \fi
334   \fi
335   \let\SX@lst@Init=\lst@Init

```

Prevents float environments from floating. This is not enough for floating listing environments! Why?

```

336   \edef\SX@save@cnt{\arabic\@capttype}%
337   \def\@xfloat##1[##2]{%
338     \def\@capttype{##1}%
339     \setcounter\@capttype{-1}%
340     \@float@HH{##1}[H]}%

```

Special handling of floating listing environments.

```

341   \def\lst@Init{%
342     \let\lst@float=\relax
343     \setcounter\@capttype{-1}%
344     \SX@lst@Init
345   }

```

Typeset the Code.

```

346   \SX@tempb{\SX@tempa{\SX@codefile}}\par

```

Restore the regular numbering of floats outside of ‘LTXexample’.

```

347   \setcounter\@capttype{\SX@save@cnt}%
348   \endgroup
349   \else
350     \expandafter\includegraphics\expandafter[\SX@graphicparam]%
351     {\SX@graphicname}%
352   \fi
353 }

```

`\SX@codeInput`

```

354 \newcommand*\SX@codeInput{%

```

Without a caption entry the command `\lstinputlisting` adds the filename to the “list of listings” (lol). This should be avoided.

```

355   \begingroup

```

The default parameters for all examples.

```

356   \expandafter\lstset\expandafter{\SX@explpreset}%

```

If “numbers=none” then margin dimensions should be zero.

```

357   \expandafter\lstset\expandafter{\SX@@explpreset}%
358   \ifx\lst@PlaceNumber\@empty
359     \g@addto@macro\SX@@explpreset{,xleftmargin=0pt,xrightmargin=0pt}%
360   \fi
361   \SX@Info
362   \expandafter\lstinputlisting\expandafter[\SX@@explpreset,nolol=true,%
363     caption={},belowskip=\z@,aboveskip=\z@,float=false]{\SX@codefile}%
364   \endgroup
365 }%

```

```

366 \newcommand*\LTxinputExample[2] [] {%
367   \g@addto@macro\SX@@explpreset{#1,codefile=#2}%
368   \SX@put@code@result}%

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

.
372 \AtBeginDocument{%
373   \def\theHlstnumber{\theHlstlisting.\arabic{lstnumber}.\lst@neglisting}%
374 }

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

```

Change History

v0.1a	General: “rangeaccept” added (RN).	3
	General: “hpos” and “vpos” added, “pos” removed (RN). . .	3
	Initial version	1
v0.1b	\SX@put@t/b/l/r/o/i: Change [a]bove to [t]op (RN).	5
	\SX@put@t/b/l/r/o/i: Positioning the captions more independend of the result and code area (RN).	5
v0.1c	\SX@put@t/b/l/r/o/i: Commands \SX@KillAboveCaptionskip and \SX@KillBelowCaptionskip added (RN).	5
v0.1f	General: “lstdescript” added. (RN). . .	3
v0.1h	General: “codefile” added. (RN). . .	3
	“lstdescript” renamed to “explpreset” (RN).	3
	New macro \LTxinputExample (RN).	11
	Renamed from “example” to “LTxexample” (RN).	6
v0.1i	General: Better caption positioning and correct distance between the parts (RN).	6
v0.1j	\SX@input: For ranges of lines (RN).	8
	General: “rangeaccept” added (RN).	3
	\SX@put@t/b/l/r/o/i: Change [a]bove to [t]op (RN).	5
	General: Some bug corrections (RN).	3
	General: “graphic” added (RN). . .	3
	General: Problem related to \label/\ref solved (RN). . . .	6
	General: “varwidth” and “justification” added (RN). . .	3
	“varwidth” package used (RN). . .	6
	General: Check if \SX@put@? is defined (RN).	6
	\SX@attachfile: Attach file functionality (with pdfTeX) added (RN).	5
	General: “attachfile” added (RN). . .	3
	\SX@resultInput: Input of result code now inside a group; \makeatother added (RN). . . .	9
	\SX@resultInput: Wrong catcode for newline char corrected (RN).	9

v0.3d	\SX@resultInput: Missing \par added (RN).	9	v0.3l	\SX@resultInput: Code for “scaled” option (RN).	9
v0.3e	\SX@@preset: More redefinitions added (RN).	4		General: Option “scaled” and \SX@scaled added (RN).	3
v0.3g	General: \SX@ProcessResult is now working correctly using \readline and \scantokens. Thanks to Ulrich Diez for help (RN).	8	v0.3m	\SX@put@code@result: Wrong assignment for \lst@belowskip (RN).	7
	Missing \newcommand for \SX@@explpreset added (RN).	4	v0.3n	\SX@put@code@result: Use \ResultBox	7
v0.3h	General: New Option ‘attachfiles’ (RN).	3		General: Define \ResultBox etc.	3
v0.3j	\SX@put@code@result: Setting \lst@MakeCaption to was a bad idea for hyperlinks. Group added to varwidth environment. (Suggestions by Ulrike Fischer.).	7	v0.3p	\SX@@preset: Remove extra treatment of ‘figure’/‘table’ (RN).	4
v0.3k	\SX@put@code@result: Setting \lst@MakeCaption to \@gobble again (prevent multiply defined labels; label key)	7		\SX@put@code@result: Let’s leave \lst@MakeCaption untouched (RN).	7
	General: Definition for “hyperref” (suggested by Heiko Oberdiek)	11		\SX@resultInput: Better handling of floats (RN).	10
				\isSX@odd: Replace \isodd with \ifodd\getpagerefnumber (remove package ‘ifthen’) (RN).	4
				General: Remove package ‘calc’ (RN).	3
			v0.3q	\SX@resultInput: Floats should always be numbered 0 (RN).	10

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	\@footnotetext	78	\abovedisplayshortskip	80
\%	\@gobble		\abovedisplayskip	79
\@@input 71–73, 75, 78, 211		\aftergroup	88, 90
\@SX@attachfilefalse	\@gobbletwo	74	\arabic	336, 373
\@SX@orangeacceptfalse	\@inputcheck 238, 240,		\attachfile	107
.	248, 260, 278, 289			
\@SX@varwidthfalse	\@latex@error	226		
\@SX@widefalse	\@latexerr	275	B	
\@addtofilelist	\@secondoftwo	96	\belowcaptionskip	154
\@capttype	\@temptokena	158, 167	\bibliography	72
338, 339, 343, 347	\@xfloat	337	\bigskipamount	371
\@ehd	\^	286, 319	\box	228
\@firstofone				
\@firstoftwo				
\@float@HH				
	A		C	
	\abovecaptionskip	148	\c@lstlisting	160

\SX@graphicparam ..	\SX@ProcessResult .	251, 254, 267,
..... 42, 45, 247, 251, 288	327, 330, 332, 346
165, 168, 311, 350	\SX@put@code@result	\SX@vsep
\SX@hsep 174, 176, 368	24, 55, 113, 118, 298
23, 54, 122, 126, 297	\SX@put@t	\SX@wasoddfalse ... 90
\SX@IDENT . 87, 104, 189	\SX@put@t/b/l/r/o/i <u>111</u>	\SX@wasoddtrue . 88, 100
\SX@Info .. 285, 293, 361	\SX@ResBox	\SX@width ... 22, 53,
\SX@input <u>269</u> , 322	. 48, 214, 224, 228	190, 191, 199,
\SX@justification .	\SX@ResultArea 112,	205, 206, 219,
..... 39, 62, 136	119, 123, 127, 135	224, 228, 296, 330
\SX@KillAboveCaptionskip	\SX@resultInput 221, <u>316</u>	T
..... 145, 213	\SX@rframe	\theHlstnumber 373
\SX@KillBelowCaptionskip	.. 27, 57, 201, 300	\thelstlisting 373
..... 151, 229	\SX@save@cnt .. 336, 347	\theltxexample 107
\SX@lines . 263, 287, 290	\SX@scaled	\thispagestyle 73
\SX@lst@Init .. 335, 344	29, 30, 326, 329, 332	U
\SX@overhang . 25, 56,	\SX@SkipToFirst ...	\usebox
187, 197, 210, 299	. 237, 243, 257, 287	142
\SX@pos . 21, 52, 194,	\SX@tempa . 240, 260,	W
196, 225–227, 295	263, 322, 324, 346	\write
\SX@preset . 28, 58, 320	\SX@tempb 249,	74