

The `showexpl` package*

Rolf Niepraschk (Rolf.Niepraschk@gmx.de)

2021/01/06

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[\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` v0.3s, dated 2021/01/06.

- 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 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 T_EX 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
72   \expandafter\ifx\csname ver@cleveref.sty\endcsname\relax
73     \let\refstepcounter=\stepcounter
74     \let\label@gobble
75   \else
76     \let\cref@old@refstepcounter=\stepcounter
77     \let\cref@old@label=\@gobble
78   \fi
79   \let\bibliography@gobble
80   \let\pagestyle@gobble \let\thispagestyle@gobble
81   %%\let\immediate\relax \let\write@gobbletwo
82   %%\let\closeout@gobble \let\@input@gobble
83   \renewcommand\marginpar[2] [] {}%
84   \renewcommand\footnote[2] [] {}%
85   \let\@footnotetext@gobble
86   %%\abovedisplayskip=\z@
87   %%\abovedisplayshortskip=\z@
88 }
89 \newcommand*\SX@eat@version[1] [] {}

```

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

```

90 \newif\ifSX@wasodd
91 \if@twoside
92   \newcommand*\isSX@odd{%
93     \begingroup
94       \ifodd\getpagerefnnumber{\SX@IDENT}}%
95     \aftergroup\SX@wasoddtrue
96   \else
97     \aftergroup\SX@wasoddfalse
98   \fi
99 \endgroup
100 \ifSX@wasodd
101   \expandafter\@firstoftwo
102 \else
103   \expandafter\@secondoftwo
104 \fi
105 }
106 \else
107   \SX@wasoddtrue
108   \newcommand*\isSX@odd[2]{#1}
109 \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.

```

110 \newcounter{ltxexample}
111 \newcommand*\SX@IDENT{\SX@number\value{ltxexample}}

```

`\SX@attachfile`

```

112 \newcommand*\SX@attachfile{%
113   \if@SX@attachfile
114     \attachfile[mimetype=text/plain,subject={example \theltxexample}]%
115     {\SX@codefile}{}%
116   \fi
117 }

```

`\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.

```

118 \newcommand*\SX@put@t[3]{%
119   \SX@ResultArea{\linewidth}{#2}\endgraf\pagebreak[2]%
120   \@tempdima=\dimexpr\SX@vsep\vskip\@tempdima
121   \SX@CodeArea{\linewidth}{#3}%
122 }
123 \newcommand*\SX@put@b[3]{%
124   \SX@CodeArea{\linewidth}{#3}\endgraf\pagebreak[2]%
125   \@tempdima=\dimexpr\SX@vsep\vskip\@tempdima
126   \SX@ResultArea{\linewidth}{#2}%
127 }
128 \newcommand*\SX@put@l[3]{%
129   \@tempdimc=\dimexpr\linewidth-#1-\SX@hsep %
130   \SX@ResultArea{#1}{#2}\hfill\SX@CodeArea{\@tempdimc}{#3}%
131 }
132 \newcommand*\SX@put@r[3]{%
133   \@tempdimc=\dimexpr\linewidth-#1-\SX@hsep %
134   \SX@CodeArea{\@tempdimc}{#3}\hfill\SX@ResultArea{#1}{#2}%

```

```

135 }
136 \newcommand*\SX@put@o[3]{%
137   \@nameuse{SX@put@\ifSX@wasodd r\else l\fi}{#1}{#2}{#3}%
138 }
139 \newcommand*\SX@put@i[3]{%
140   \@nameuse{SX@put@\ifSX@wasodd l\else r\fi}{#1}{#2}{#3}%
141 }
142 \newcommand\SX@ResultArea[2]{%
143   \SX@justification\@tempdima=\dimexpr #1 %
144   \parbox\@tempdima{#2}%
145 }
146 \newcommand\SX@CodeArea[2]{%
147   \@tempdima=\dimexpr #1 %
148   \sbox\@tempboxa{\parbox\@tempdima{#2}}%
149   \@tempdima=\dp\@tempboxa\usebox\@tempboxa
150   \rlap{\raisebox{-\@tempdima}[Opt][Opt]{\SX@attachfile}}%
151 }
152 \newcommand*\SX@KillAboveCaptionskip{%
153   \ifx\lst@caption\@empty\else
154     \lst@ifsubstring t\lst@captionpos
155     {\vskip-\abovecaptionskip}{}%
156   \fi
157 }
158 \newcommand*\SX@KillBelowCaptionskip{%
159   \ifx\lst@caption\@empty\else
160     \lst@ifsubstring b\lst@captionpos
161     {\vskip-\belowcaptionskip}{}%
162   \fi
163 }

```

LTXexample

```

164 \lstnewenvironment{LTXexample}[1] []{%
165   \@temptokena{#1}%
166   \begingroup
   For "codefile=..."/"graphic=..." if \theltxexample or \thelstlisting is part of
   the filename.
167   \advance\c@ltxexample\@ne \advance\c@lstlisting\@ne
168   \expandafter\lstset\expandafter{\SX@explpreset,#1}%
169   \edef\x{\endgroup
170     \def\noexpand\SX@codefile{\SX@codefile}%
171     \def\noexpand\SX@graphicname{\SX@graphicname}%
172     \def\noexpand\SX@graphicparam{\SX@graphicparam}}%
173   \x
174   \xdef\SX@@explpreset{\the\@temptokena,codefile=\SX@codefile,%
175     graphic={[\SX@graphicparam]{\SX@graphicname}}}%
176   \setbox\@tempboxa=\hbox\bgroup
177   \lst@BeginWriteFile{\SX@codefile}%
178 }
179 {%
180   \lst@EndWriteFile\egroup
181   \SX@put@code@result
182 }

```

\SX@put@code@result

```
183 \newcommand*\SX@put@code@result{%
184   \begingroup
185     \expandafter\lstset\expandafter{\SX@explpreset}%
186     \expandafter\lstset\expandafter{\SX@@explpreset}%
    Use listings floating procedure if necessary.
187     \ifx\lst@float\relax\else
188       \edef\@tempa{\noexpand\lst@beginfloat{lstlisting}[\lst@float]}
189       \expandafter\@tempa
190     \fi
191     \ifx\lst@caption\@empty
192       \lstset{lolol=true}%
193     \fi
194     \if@SX@wide\def\SX@overhang{\marginparwidth+\marginparsep}\fi
195     \trivlist\item\relax
196     \stepcounter{ltxexample}\label{\SX@IDENT}%
    Make \SX@width a real dimension if the unit is missing.
197     \SX@defaultWD\SX@width{\SX@width}%
    Set the default width if necessary.
198     \ifdim\SX@width<\z@
199       \@tempswatru
200       \def\@tempa{t}%
201       \ifx\@tempa\SX@pos\@tempswafalse\fi
202       \def\@tempa{b}%
203       \ifx\@tempa\SX@pos\@tempswafalse\fi
204       \@tempdima=\dimexpr\linewidth+\SX@overhang %
205       \if@tempswa\@tempdima=.5\@tempdima\fi%
206       \edef\SX@width{\the\@tempdima}%
207     \fi
    Correct \SX@width if a frame is requested.
208     \ifx\SX@rframe\@empty
209       \long\def\SX@frame##1{##1}%
210     \else
211       \let\SX@frame\ResultBox
212       \@tempdima=\dimexpr\SX@width-2\ResultBoxSep-2\ResultBoxRule %
213       \edef\SX@width{\the\@tempdima}%
214     \fi
215     \isSX@odd{\def\@tempa{l}}{\def\@tempa{r}}%
216     \makebox[\linewidth][\@tempa]{%
217       \parbox{\dimexpr\linewidth+\SX@overhang}{%
\SX@codefile (\jobname.tmp) is not nessesary for the filelist.
218       \let\@addtofilelist\@gobble
219       \let\lst@ifdisplaystyle=\iftrue
220       \SX@KillAboveCaptionskip\lst@MakeCaption{t}%
    Use the “natural” width of the result code if “varwidth” is true. .
221       \setbox\SX@ResBox\hbox{%
222         \fboxsep=\ResultBoxSep
223         \fboxrule=\ResultBoxRule
224         \SX@frame{%
225           \@nameuse{\if@SX@varwidth varwidth\else minipage\fi}%
```

```

226         \SX@width\relax
227         \begingroup
228         \SX@resultInput
229         \endgroup
230         \@nameuse{end\if@SX@varwidth varwidth\else minipage\fi}}}%
231     \edef\SX@width{\the\wd\SX@ResBox}%
232     \@ifundefined{SX@put@\SX@pos}%
233     {\@latex@error{Parameter ‘\SX@pos’ undefined}\@ehd}%
234     {\@nameuse{SX@put@\SX@pos}%
235     {\SX@width}{\box\SX@ResBox}{\SX@codeInput}}}%
236     \lst@MakeCaption{b}\SX@KillBelowCaptionskip
237     }%
238 }%
239 \endtrivlist
240 \ifx\lst@float\relax\else\expandafter\lst@endfloat\fi
241 \gdef\SX@explpreset{}%
242 \endgroup
243 }

244 \newcommand\SX@SkipToFirst{%
245   \ifeof\@inputcheck\else
246     \ifnum \lst@lineno=\lst@firstline\else
247       \readline\@inputcheck to\SX@tempa
248       \typeout{IGNORE (\the\lst@lineno)}%
249       \global\advance\lst@lineno\@ne
250       \SX@SkipToFirst
251     \fi
252   \fi
253 }

254 \newcommand\SX@ProcessResult{%
255   \ifeof\@inputcheck
256     \let\SX@tempb\relax
257   \else
258     \let\SX@tempb\SX@ProcessResult
259     \ifnum \lst@lineno>\lst@lastline\relax
260       \ifx\lst@linerange\@empty
261         \let\SX@tempb\relax
262       \else
263         \lst@GetLineInterval
264         \SX@SkipToFirst
265       \fi
266     \else
267       \readline\@inputcheck to\SX@tempa
268       \typeout{READ (\the\lst@lineno)}%
269       \expandafter\g@addto@macro
270       \expandafter\SX@lines\expandafter{\SX@tempa^^J}%
271       \global\advance\lst@lineno\@ne
272     \fi
273   \fi
274   \SX@tempb
275 }

```

\SX@input

```

276 \newcommand\SX@input[1]{%

```



```

277 \begingroup
278 \IfFileExists{#1}{}%
279 {%
280 \filename@parse{#1}%
281 \ifx\filename@ext\relax \def\filename@ext{tex}\fi
282 \latexerr{File
283 '\filename@area\filename@base.\filename@ext' not found.^^J^^J}\@ehd%
284 }%
285 \openin\@inputcheck#1
286 \lsthk@PreSet\let\lst@linerange\@empty\global\lst@lineno\@ne
287 \expandafter\lstset\expandafter{\SX@@explpreset}%
288 \ifx\lst@linerange\@empty
289 \edef\lst@linerange{{\lst@firstline}-{\lst@lastline}},}%
290 \fi
291 \lst@GetLineInterval
292 \SX@Info
293 \newlinechar='^^J\relax
294 \SX@SkipToFirst\let\SX@lines\@empty
295 \SX@ProcessResult
296 \closein\@inputcheck
297 \scantokens\expandafter{\SX@lines}%
298 \endgroup
299 }

300 \newcommand*\SX@Info{%
301 \typeout{-----}%
302 \typeout{pos=\SX@pos}%
303 \typeout{width=\SX@width}%
304 \typeout{hsep=\SX@hsep}%
305 \typeout{vsep=\SX@vsep}%
306 \typeout{overhang=\SX@overhang}%
307 \typeout{rframe=\SX@rframe}%
308 \typeout{codefile=\SX@codefile}%
309 \@ifundefined{lst@firstline}{}%
310 {\typeout{\string\lst@firstline=\lst@firstline}}%
311 \@ifundefined{lst@lastline}{}%
312 {\typeout{\string\lst@lastline=\lst@lastline}}%
313 \@ifundefined{lst@linerange}{}%
314 {\typeout{\string\lst@linerange=\lst@linerange}}%
315 \typeout{\string\if@SX@wide=\if@SX@wide TRUE\else FALSE\fi}%
316 \typeout{\string\if@SX@rangeaccept=\if@SX@rangeaccept TRUE\else FALSE\fi}%
317 \typeout{\string\if@SX@varwidth=\if@SX@varwidth TRUE\else FALSE\fi}%
318 \typeout{graphicfile=\SX@graphicname, graphicparameter=[\SX@graphicparam]}%
319 \typeout{-----}%
320 }
321 \providecommand*\MakePercentIgnore{\catcode'\%9\relax}
322 \providecommand*\MakePercentComment{\catcode'\%14\relax}

\SX@resultInput

323 \newcommand*\SX@resultInput{%
324 \ifx\SX@graphicname\@empty
325 \begingroup
326 \MakePercentComment\makeatother\catcode'\^^M=5\relax
327 \SX@@preset\SX@preset

```

```

328     \if@SX@rangeaccept
329     \let\SX@tempa=\SX@input
330     \else
331     \let\SX@tempa=\input
332     \fi
333     \if\SX@scaled ?%
334     \let\SX@tempb=\@firstofone
335     \else
336     \if\SX@scaled !%
337     \def\SX@tempb##1{\resizebox{\SX@width}{!}{##1}}%
338     \else
339     \def\SX@tempb##1{\scalebox{\SX@scaled}{##1}}%
340     \fi
341     \fi
342     \let\SX@lst@Init=\lst@Init

```

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

```

343     \def\@xfloat##1[##2]{%
344     \def\@capttype{##1}%
345     \@namedef{the\@capttype}{0}%
346     \@float@HH{##1}[H]}%

```

Special handling of floating listing environments.

```

347     \def\lst@Init{%
348     \let\lst@float=\relax
349     \setcounter\@capttype{-1}%
350     \SX@lst@Init
351     }

```

Typeset the Code.

```

352     \SX@tempb{\SX@tempa{\SX@codefile}}\par
353     \endgroup
354     \else
355     \expandafter\includegraphics\expandafter[\SX@graphicparam]%
356     {\SX@graphicname}%
357     \fi
358 }

```

`\SX@codeInput`

```

359 \newcommand*\SX@codeInput{%

```

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

```

360 \beginingroup

```

The default parameters for all examples.

```

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

```

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

```

362 \expandafter\lstset\expandafter{\SX@@explpreset}%
363 \ifx\lst@PlaceNumber\@empty
364 \g@addto@macro\SX@@explpreset{,xleftmargin=0pt,xrightmargin=0pt}%
365 \fi
366 \SX@Info
367 \expandafter\lstinputlisting\expandafter[\SX@@explpreset,nolol=true,%

```

```

368     caption={},belowskip=\z@,aboveskip=\z@,float=false]{\SX@codefile}%
369 \endgroup
370 }%

371 \newcommand*\LTxinputExample[2] []{%
372   \g@addto@macro\SX@@explpreset{float=false,#1,codefile=#2}%
373   \SX@put@code@result}%

All the default values.
374 \lstset{explpreset={numbers=left,numberstyle=\tiny,numbersep=.3em,
Negative width means defaults.
375 xleftmargin=1em,columns=flexible,language=[LaTeX]TEX},pos=1,width=-99pt,
376 overhang=0pt,hsep=\columnsep,vsep=\bigskipamount,rframe=single}
.
377 \AtBeginDocument{%
378   \def\theHlstnumber{\thelstlisting.\arabic{lstnumber}.\lst@neglisting}%
379 }

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

```

Change History

v0.1a	the parts (RN).	6
General: “hpos” and “vpos”		
added, “pos” removed (RN). . .	3	
Initial version	1	
v0.1b		
\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: “lstpreset” added. (RN).	3	
v0.1h		
General: “codefile” added. (RN). .	3	
“lstpreset” 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		
v0.1j		
\SX@input: For ranges of lines		
(RN).	8	
General: “rangeaccept” added		
(RN).	3	
v0.1k		
\SX@put@t/b/l/r/o/i: Change		
[a]bove to [t]op (RN).	5	
General: Some bug corrections		
(RN).	3	
v0.1l		
General: “graphic” added (RN). . .	3	
v0.1m		
General: Problem related to		
\label/\ref solved (RN).	6	
v0.2a		
General: “varwidth” and		
“justification” added (RN). . . .	3	
“varwidth” package used (RN). . .	6	
v0.2b		
General: Check if \SX@put@? is		
defined (RN).	6	
v0.3a		
\SX@attachfile: Attach file		
functionality (with pdfTeX)		
added (RN).	5	
General: “attachfile” added (RN).	3	

v0.3b	<code>\SX@resultInput</code> : Input of result code now inside a group; <code>\makeatother</code> added (RN). 9	v0.3l	<code>\SX@resultInput</code> : Code for “scaled” option (RN). 9
v0.3c	<code>\SX@resultInput</code> : Wrong catcode for newline char corrected (RN). 9		General: Option “scaled” and <code>\SX@scaled</code> added (RN). 3
v0.3d	<code>\SX@resultInput</code> : Missing <code>\par</code> added (RN). 9	v0.3m	<code>\SX@put@code@result</code> : Wrong assignment for <code>\lst@belowskip</code> (RN). 7
v0.3e	<code>\SX@@preset</code> : More redefinitions added (RN). 4	v0.3n	<code>\SX@put@code@result</code> : Use <code>\ResultBox</code> 7
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		General: Define <code>\ResultBox</code> etc. . . 3
	Missing <code>\newcommand</code> for <code>\SX@@explpreset</code> added (RN). 4	v0.3p	<code>\SX@@preset</code> : Remove extra treatment of ‘figure’/‘table’ (RN). 4
v0.3h	General: New Option ‘attachfiles’ (RN). 3		<code>\SX@put@code@result</code> : Let’s leave <code>\lst@MakeCaption</code> untouched (RN). 7
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> environment. (Suggestions by Ulrike Fischer.). 7		<code>\SX@resultInput</code> : Better handling of floats (RN). 10
v0.3k	<code>\SX@put@code@result</code> : Setting <code>\lst@MakeCaption</code> to <code>\@gobble</code> again (prevent multiply defined labels; label key) 7		<code>\isSX@odd</code> : Replace <code>\isodd</code> with <code>\ifodd\getpagerefnumber</code> (remove package ‘ifthen’) (RN). 4
	General: Definition for “hyperref” (suggested by Heiko Oberdiek) 11		General: Remove package ‘calc’ (RN). 3
		v0.3q	<code>\SX@resultInput</code> : Floats should always be numbered 0 (RN). . . 10
		v0.3r	<code>\SX@@preset</code> : letting <code>\refstepcounter</code> be <code>\stepcounter</code> (RN). 4
		v0.3s	<code>\SX@@preset</code> : Special handling of <code>cleveref</code> ’s <code>\refstepcounter</code> (RN). 4

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	<code>\SX@widerfalse</code> 35	<code>\@footnotetext</code> 85
<code>\%</code> 321, 322	<code>\@addtofilelist</code> . . . 218	<code>\@gobble</code> . 71, 74, 77,
<code>\@@input</code> 82	<code>\@capttype</code> . 344, 345, 349	79, 80, 82, 85, 218
<code>\@SX@attachfilefalse</code> 36	<code>\@ehd</code> 233, 283	<code>\@gobbletwo</code> 81
<code>\@SX@rangeacceptfalse</code>	<code>\@firstofone</code> 334	<code>\@inputcheck</code> 245, 247,
. 33	<code>\@firstoftwo</code> 101	255, 267, 285, 296
<code>\@SX@varwidthfalse</code> . 34	<code>\@float@HH</code> 346	<code>\@latex@error</code> 233

<code>\setcounter</code>	349	<code>\SX@hsep</code>		<code>\SX@rframe</code>	
<code>\stepcounter</code> 73, 76, 196		23, 54, 129, 133, 304		.. 27, 57, 208, 307	
<code>\string</code> 310, 312, 314–317		<code>\SX@IDENT</code> . 94, 111, 196		<code>\SX@scaled</code>	
<code>\SX@@explpreset</code> . . .		<code>\SX@Info</code> .. 292, 300, 366		29, 30, 333, 336, 339	
..... 60, 174,		<code>\SX@input</code>	276, 329	<code>\SX@SkipToFirst</code> . . .	
186, 241, 287,		<code>\SX@justification</code> .		. 244, 250, 264, 294	
362, 364, 367, 372	 39, 62, 143		<code>\SX@tempa</code> . 247, 267,	
<code>\SX@preset</code>	63, 327	<code>\SX@KillAboveCaptionskip</code>		270, 329, 331, 352	
<code>\SX@attachfile</code> 152, 220		<code>\SX@tempb</code>	256,
..... 11, 112, 150		<code>\SX@KillBelowCaptionskip</code>		258, 261, 274,	
<code>\SX@CodeArea</code> .. 121,	 158, 236		334, 337, 339, 352	
124, 130, 134, 146		<code>\SX@lines</code> . 270, 294, 297		<code>\SX@vsep</code>	
<code>\SX@codefile</code> . 32, 61,		<code>\SX@lst@Init</code> .. 342, 350		24, 55, 120, 125, 305	
115, 170, 174,		<code>\SX@overhang</code> . 25, 56,		<code>\SX@wasoddfalse</code>	97
177, 308, 352, 368		194, 204, 217, 306		<code>\SX@wasoddtrue</code> . 95, 107	
<code>\SX@codeInput</code> . 235, <u>359</u>		<code>\SX@pos</code> . 21, 52, 201,		<code>\SX@width</code> . . . 22, 53,	
<code>\SX@def@WD</code>	18–20	203, 232–234, 302		197, 198, 206,	
<code>\SX@defaultWD</code> .. <u>17</u> , 197		<code>\SX@preset</code> . 28, 58, 327		212, 213, 226,	
<code>\SX@eat@version</code> . . .		<code>\SX@ProcessResult</code> .		231, 235, 303, 337	
..... 64, 65, 89	 254, 258, 295			
<code>\SX@explpreset</code>		<code>\SX@put@code@result</code>		T	
31, 59, 168, 185, 361	 181, <u>183</u> , 373		<code>\theHlstnumber</code>	378
<code>\SX@frame</code> . 209, 211, 224		<code>\SX@put@t</code>	118	<code>\thelstlisting</code>	378
<code>\SX@graphicname</code> . . .		<code>\SX@put@t/b/l/r/o/i</code> <u>118</u>		<code>\theltxexample</code>	114
... 41, 45, 171,		<code>\SX@ResBox</code>		<code>\thispagestyle</code>	80
175, 318, 324, 356		. 48, 221, 231, 235			
<code>\SX@graphicparam</code> . .		<code>\SX@ResultArea</code> 119,		U	
..... 42, 45,		126, 130, 134, 142		<code>\usebox</code>	149
172, 175, 318, 355		<code>\SX@resultInput</code> 228, <u>323</u>			
				W	
				<code>\write</code>	81