fitr: Creating FitR Destinations

D. P. Story Email: dpstory@acrotex.net

processed July 9, 2020

Contents

1	Options and Required Packages	2
2	The Main Code	4
3	Document JavaScript	12
4	Index	16
	Change History (*package)	18

Introduction. This package supports dvips and dvipsone utilities, and assumes the PDF creator is **Adobe Distiller**. It also supports the pdflatex (and lualatex) executable through the pdftex option; the options dvipdfm, dvipdfmx, and xetex support the corresponding executables.

When using a dvi-ps-pdf work flow (with Distiller as the PDF creator), the package uses the **pdfmark** for the FitR view-type destination:

```
[ /Dest/\(\langle name \rangle \)/View [ /FitR left bottom right top ]/DEST pdfmark
```

For the pdftex or luatex option, we use \pdfdest a primitive of that utility:

```
\begin{array}{ll} \texttt{\phace points} & \texttt{\phace points} & \texttt{\phace fitr} \\ & \texttt{\phace width} & \texttt{\phace width} & \texttt{\phace width} \\ & \texttt{\phace height} & \texttt{\phace height} & \texttt{\phace depth} \\ & \texttt{\phace depth} & \texttt{\phace depth} & \texttt{\phace depth} \end{array}
```

For the dvipdfm, dvipdfmx, and xetex options, the problem of computing the FitR rectangle is a little trickier.

where $\mathtt{Qurx}\langle\mathtt{name}\rangle$ and $\mathtt{Qury}\langle\mathtt{name}\rangle$ are the calculated upper right x and y coordinate of the bounding box.

The PDF Reference describes FitR as,

Display the page designated by page, with its contents magnified just enough to fit the rectangle specified by the coordinates *left*, *bottom*, *right*, and *top* entirely within the window both horizontally and vertically.

This type of destination allows for the creation of a link or form button to zoom in on a rectangular region.

1 Options and Required Packages

- 2 \RequirePackage{xkeyval}
- 3 \RequirePackage{ifpdf}[2006/02/20]
- 4 \RequirePackage{ifxetex}[2006/08/21]

(2020/07/03) Added ifluatex package for later testing

5 \RequirePackage{ifluatex}

dvipsone
pdftex
luatex
dvipdfm
dvipdfmx
xetex

Driver options. Three options, dvips (the default) and dvipsone, both of which require **Adobe Distiller** as the PDF creator; the pdflatex and luatex allows for the same functionality for the pdflatex executable. We also support dvipdfm and its extensions, dvipdfmx and xetex.

Auto detection of drivers. The drivers pdflatex, lualatex, and xelatex are automatically detected, so there is no need to specify their respective options: pdftex, luatex, and xetex.

preview

viewMagWin

Preview options. The preview option is a carry over from eforms. When selected, all form fields are outlined; useful when setting the location of fields in a dvi previewer. The other option is viewMagWin will show the viewing windows surrounding the target. This is the rectangle that will be jumped to. Use this option to adjust the size of the window to your needs. When either of the last two options has an exclamation point prior, that means to turn off the switch. You can conveniently use viewMagWin to see the viewing window, then change it to !viewMagWin to remove the visible window. Cool. Similarly, you can turn off preview using !preview.

!viewMagWin !preview

We create \ifpreview and \ifviewMagWin, and declare the options.

- 6 \@ifundefined{ifpreview}{\newif\ifpreview \previewfalse}{}
- 8 \let\fitr@driver\@empty

The pdftex option

- 9 \DeclareOptionX{pdftex}{\gdef\fitr@driver{pdftex}%
- 10 \PassOptionsToPackage{\fitr@driver}{hyperref}%
- 11 \PassOptionsToPackage{\fitr@driver}{eforms}}
- 12 \def\fitr@pdftex@driver{pdftex}

```
The luatex option
                13 \DeclareOptionX{luatex}{\gdef\fitr@driver{luatex}%
                    \PassOptionsToPackage{\fitr@driver}{hyperref}%
                     \PassOptionsToPackage{\fitr@driver}{eforms}}
                16 \def\fitr@pdftex@driver{luatex}
                Distiller based drivers.
                17 \DeclareOptionX{dvips}{\gdef\fitr@driver{dvips}%
                     \PassOptionsToPackage{\fitr@driver}{hyperref}%
                    \PassOptionsToPackage{\fitr@driver}{eforms}}
                20 \def\fitr@dvips@driver{dvips}
                21 \DeclareOptionX{dvipsone}{\gdef\fitr@driver{dvipsone}%
                     \PassOptionsToPackage{\fitr@driver}{hyperref}%
                     \PassOptionsToPackage{\fitr@driver}{eforms}}
                24 \def\fitr@dvipsone@driver{dvipsone}
                dvipdfm and its variants.
                25 \newif \if@fitr@dvipdfm \@fitr@dvipdfmfalse
                26 \DeclareOptionX{dvipdfm}{\gdef\fitr@driver{dvipdfm}%
                    \@fitr@dvipdfmtrue
                     \PassOptionsToPackage{\fitr@driver}{hyperref}%
                28
                    \PassOptionsToPackage{\fitr@driver}{eforms}}
                30 \def\fitr@dvipdfm@driver{dvipdfm}
                31 \DeclareOptionX{dvipdfmx}{\gdef\fitr@driver{dvipdfmx}%
                    \@fitr@dvipdfmtrue
                    \PassOptionsToPackage{\fitr@driver}{hyperref}%
                33
                    \PassOptionsToPackage{\fitr@driver}{eforms}}
                34
                35 \def\fitr@dvipdfmx@driver{dvipdfmx}
                36 \DeclareOptionX{xetex}{\gdef\fitr@driver{xetex}%
                37
                    \@fitr@dvipdfmtrue
                    \PassOptionsToPackage{\fitr@driver}{hyperref}%
                    \PassOptionsToPackage{\fitr@driver}{eforms}}
                40 \def\fitr@xetex@driver{xetex}
                Various preview options.
      preview
      !preview
                41 \DeclareOptionX{preview}{\previewtrue}
   viewMagWin
                42 \DeclareOptionX{!preview}{\previewfalse}
   !viewMagWin
                43 \DeclareOptionX{viewMagWin}{\viewMagWintrue}
                44 \DeclareOptionX{!viewMagWin}{\viewMagWinfalse}
   \previewOn
                Some convenience macros to locally turn on or off preview and viewMagWin.
   \previewOff
                45 \providecommand{\previewOn}{\previewtrue}
\viewMagWinOn
                46 \providecommand{\previewOff}{\previewfalse}
\viewMagWinOff
                47 \providecommand{\viewMagWinOn}{\viewMagWintrue}
                48 \providecommand{\viewMagWinOff}{\viewMagWinfalse}
                When the gonative option is specified, field and document JavaScript are not
     gonative
                used, as a result, no special effects are implemented. This gives a basic document
                that can be compiled successfully with the dvips -> ps2pdf workflow.
                49 \newif\ifFRusedljs \FRusedljstrue
                50 \DeclareOptionX{gonative}{\FRusedljsfalse}
```

```
blinkonjmp
                 Target region blinks on jump
   !blinkonjmp
                 51 \newif\ifFRblinkonjmp \FRblinkonjmpfalse
                 52 \DeclareOptionX{blinkonjmp}{\FRblinkonjmptrue}
                 53 \DeclareOptionX{!blinkonjmp}{\FRblinkonjmpfalse}
blinkonrestore
                 Target region blinks on restore
!blinkonrestore
                 54 \neq FRblinkonrestore FRblinkonrestorefalse
                 55 \DeclareOptionX{blinkonrestore}{\FRblinkonrestoretrue}
                 56 \DeclareOptionX{!blinkonrestore}{\FRblinkonrestorefalse}
                A convenience option, blink is equivalent to blinkonjmp and blinkonrestore.
         blink
         !blink
                 57 \DeclareOptionX{blink}{\FRblinkonjmptrue\FRblinkonrestoretrue}
                 58 \DeclareOptionX{!blink}{\FRblinkonjmpfalse\FRblinkonrestorefalse}
                 Before processing options, try for automatic driver detection.
                 59 \@ifpackageloaded{web}{\ExecuteOptionsX{\eq@driver@name}}{%
                     \ifluatex\ExecuteOptionsX{pdftex}\else
                     \ifpdf\ExecuteOptionsX{pdftex}\else
                 61
                     \ifxetex\ExecuteOptionsX{xetex}\else
                 62
                     \@ifundefined{l@tex@@@@driver}{\ExecuteOptionsX{dvips}}
                 63
                         {\ExecuteOptionsX{dvipsone}}\fi\fi\fi
                 64
                 65 }
                 Process the options
                 66 \ProcessOptionsX
```

Required packages

67 \RequirePackage{xcolor}

Minimal packages calc and eforms (which inputs hyperref).

- 68 %\RequirePackage{hyperref}
- 69 \RequirePackage{eforms}[2020/07/05]
- 70 \RequirePackage{calc}

We also require collectbox, a cool way of collecting an argument which enables the use of verbatim text. Written by Martin Scharrer.

71 \RequirePackage{collectbox}

2 The Main Code

Some scratch counters, lengths, boxes.

- 72 \newcounter{magCnt}
- 73 \newbox\fitr@bbox
- 74 \newcount\fitr@height
- $75 \newcount\fitr@width$
- 76 \newcount\fitr@depth
- 77 \newlength\fitr@length

For xelatex we adjust field sizes to conform to the same sizes produced by pdflatex and latex. For lualatex, make a definition so it can be process just like pdflatex.

```
78 \ifxetex\makeXasPDOff\fi
                 79 \ifluatex\protected\def\pdfdest{\pdfextension dest }\fi
\get@fitr@dimen{\(content\)\} Tthis command encloses \(\lambda content\)\) in the box register \fitr@bbox
                 (an \hbox) and transfers dimension info to the count registers \fitr@height,
                  \fitr@width, and \fitr@depth. To typeset \(\langle content \rangle \), \unbox\fitr@bbox.
                 80 \def\get@fitr@dimen#1{%
                      \setbox\fitr@bbox=\hbox{#1}%
                      \fitr@height=\ht\fitr@bbox
                        \xdef\fitr@height@l{\the\ht\fitr@bbox}%
                 83
                      \fitr@width=\wd\fitr@bbox
                 84
                        \xdef\fitr@width@l{\the\wd\fitr@bbox}%
                 85
                      \fitr@depth=\dp\fitr@bbox
                 86
                        \xdef\fitr@depth@l{\the\dp\fitr@bbox}%
                 87
                 88
                      \setlength\fitr@length{\ht\fitr@bbox+\dp\fitr@bbox}%
                 89
                      \edef\fitr@@height{\the\fitr@length}%
                 dvips. The following commands are some PostScript for the dvips executable.
    \mathbf{FitRbboxB}\{\langle amt-wd\rangle\}\{\langle amt-ht\rangle\}
                 91 \def\fitr@urxury@fixup#1#2#3{}
                 92 \ifx\fitr@driver\fitr@dvips@driver
                 93 \headerps@out{/TeXtoPDF {65536 div DVImag mul} def % sp to pts
                 94 /SPtoDvips{TeXtoPDF PDFToDvips} def} % sp to dots
                 95 \def\FitRbboxB#1#2{% Uses \fitr@bbox
                      currentpoint 2 copy DvipsToPDF \the\fitr@depth\space TeXtoPDF add
                      neg vsize add 72 sub #2\space sub exch
                     DvipsToPDF 72 add #1\space sub exch
                                                                         % x1
                 99 4 2 roll exch DvipsToPDF \the\fitr@width\space
                 100 TeXtoPDF add 72 add #1\space add exch
                                                                         % x2
                 101 DvipsToPDF \the\fitr@height \space TeXtoPDF sub
                                                                         % y2
                 102 neg vsize add 72 sub #2\space add}
                 103 \else
                 104 \ifx\fitr@driver\fitr@dvipsone@driver
                 dvipsone. Definition of \FitRbbox for dvipsone.
                 105 \special{!/TeXtoPDF {65536 div mag 1000 div mul} def
                 106 / PDFtoTeX {65536 mul mag 1000 div div} def}
                 107 \def\FitRbboxB#1#2{%
                       currentpoint 2 copy \the\fitr@depth\space add DvipsToPDF
                108
                       neg PageHeight add 72 sub #2\space sub
                109
                                                                      % y1
                       exch DvipsToPDF 72 add #1\space sub exch
                                                                      % x1
                 110
                       4 2 roll exch \the\fitr@width\space add
                111
                       DvipsToPDF 72 add #1\space add exch
                                                                      % x2
                112
                       \the\fitr@height\space sub DvipsToPDF neg
                113
                       PageHeight add 72 sub #2\space add}
                                                                      % y2
                114
                115 \else\ifpdf
```

pdftex. Definition of \FitRbbox for pdftex and luatex. Uses \fitr@bbox; #1 is amount to widen box; #2 amount to heighten box.

This version of \FitRbbox expands the bounding rectangle to the viewing window by increasing the dimensions of width, height and depth. It saves these values in the macros \fitr@pdftex@view@width|height|depth.

```
116 \def\FitRbboxB#1#2{%
117
     \fitr@length=#1bp
118
     \fitr@length=2\fitr@length
     \advance\fitr@length\wd\fitr@bbox
119
120
     \edef\fitr@pdftex@view@width{\the\fitr@length}%
     \fitr@length=#2bp
121
     \advance\fitr@length\ht\fitr@bbox
122
     \edef\fitr@pdftex@view@height{\the\fitr@length}%
123
     \fitr@length=#2bp
124
     \advance\fitr@length\dp\fitr@bbox
125
     \edef\fitr@pdftex@view@depth{\the\fitr@length}%
126
127 }
128 \else\if@fitr@dvipdfm
dvipdfm/dvipdfmx/xetex. The primitive \special{pdf: dest (name)...}
is used along with the primitive variables @xpos and @ypos. The third argument
is \langle name \rangle.
129 \def\FitRbboxB#1#2#3{%
     \smash{\raisebox{-\fitr@depth@l-#2bp}%
130
       {\makebox[0pt][1]{\hspace*{-#1bp}%
131
The values of @xpos and @ypos correspond to the coordinates of the lower left
corner. The values of @urx#3 and @ury#3 are calculated in \fitr@urxury@fixup
as the coordinates of the upper right corner.
     \@pdfm@mark{dest (#3) %
       [ Othispage /FitR Oxpos Oypos Ourx#3\space Oury#3\space ]}}}}
\fitr@urxury@fixup is placed just after \fitr@bbox.
134 \def\fitr@urxury@fixup#1#2#3{%
     \smash{\raisebox{\fitr@height@1+#2bp}%
     {\makebox[0pt][1]{\hspace*{#1bp}%
136
We raise up to the upper right corner of the content box and define @urx#3 and
Oury#3 as the current values of Oxpos and Oypos,
     \@pdfm@mark{obj @urx#3\space @xpos}%
138
     \@pdfm@mark{obj @ury#3\space @ypos}}}}%
139 }
140 \fi\fi\fi\fi
We use a short macro from graphics package. We do not require the graphics
```

command so we redefine it under a different name.

```
141 \def\fitr@defaultbp#1#2{%
     \afterassignment\fitr@def@bp\dimen@#2bp\relax{#1}{#2}}
143 \def\fitr@def@bp#1\relax#2#3{%
   \if!#1!%
```

```
\def#2{#3}%
                  145
                             \else
                  146
                                  \dimen@.99626\dimen@
                  147
                                  \edef#2{\strip@pt\dimen@}%
                  148
                             \fi}
                  149
                    The major command of this package is \jdRect. Its second required argument has
                    several key-value pairs that are recognized. Options for the \jdRect command.
        lift is the amount of lift (performed by \raisebox. The value can be a positive or
                    negative length. Positive to translate upward, negative to translate downward.
                    The default is Opt.
                   150 \define@key{fitr}{lift}[0pt]{{\setlength\@tempdima{#1}%
                             \xdef\fitr@temp@length{\the\@tempdima}}%
                   152
                             \edef\fitr@@lift{\fitr@temp@length}}
                   153 \def\fitr@@lift{0pt}
      shift is the amount of horizontal shift, positive to the right, negative to the left. The
                    default is Opt
                   154 \define@key{fitr}{shift}[0pt]{{\setlength\@tempdima{#1}%
                             \xdef\fitr@temp@length{\the\@tempdima}}%
                             \edef\fitr@@shift{\fitr@temp@length}}
                   157 \def\fitr@@shift{Opt}
                   are the dimensions of the bounding box for the button/view rectangle. These
                    dimensions are ignored when the first optional parameter of \jdRect is nonempty.
   height
                   158 \define@key{fitr}{width}[0pt]{{\setlength\@tempdima{#1}%
                             \xdef\fitr@temp@length{\the\@tempdima}}%
                  159
                             \edef\fitr@@width{\fitr@temp@length}}
                  160
                   161 \def\fitr@@width{Opt}
                   162 \end{fitr} {height} [0pt] {{\tt setlength} @tempdima{\#1}\%}
                             \xdef\fitr@temp@length{\the\@tempdima}}%
                   164
                             \edef\fitr@@height{\fitr@temp@length}}
                   165 \def\fitr@@height{Opt}
                  is the reference point. Values are t (the default), c center, and b bottom. When
                    the optional first parameter of \ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\ightrightarrow\igh
                   166 \define@choicekey+{fitr}{ref}[\val\nr]{t,c,b}[t]%
                             {\edef\fitr@@refPt{\val}}{}
                   168 \def\fitr@@refPt{t}
                    add additional length around the (internal) box. This larger rectangle is the view
adddestw
adddesth
                    window.
                  169 \define@key{fitr}{adddestw}[0]{%
                             \fitr@defaultbp{\fitr@@adddestw}{#1}}
                  171 \def\fitr@@adddestw{0}
                   172 \define@key{fitr}{adddesth}[0]{%
                             \fitr@defaultbp{\fitr@@adddesth}{#1}}
                   174 \def\fitr@@adddesth{0}
   button is a Boolean. If true (the default), \jdRect creates a push button around the
                    content. When the user pushes the button, the viewer zooms in to the view
```

window. Clicking again restores the previous view. When button is false, the button is not created, but the viewing window is. You can then jump to the viewing window with a separate link or button.

```
175 \define@boolkey{fitr}{button}[true]{}
176 \KV@fitr@buttontrue
```

link is an key, which if taken, a link rather than a push button, is created. The link key takes one of two values: jmp, restore.

- When link=jmp, a link is created that jumps to the destination determined by dest. This value is designed for external links for jumping to a rectangular region Use the key nodest to keep \jdRect from creating a destination at the link location.
- When link=restore, again, a link is created around the content, a restore last view action is associated with the link; clicking on it jumps back to the view at the time you jumped.

nodest is a Boolean, which if true, causes \jdRect not to generate a destination at the current link (or button).

180 \define@boolkey{fitr}{nodest}[true]{}

181 \KV@fitr@nodestfalse

dest When button=true (the default), the name of the destination is automatically generated, internally. When button=false, you need to specify a destination to reference in a separate link or push button.

allowfX A Boolean switch (of sorts). The fitr package allows for special effects (FX) when a viewing window is jumped to and when the view is restored. The default value of allowfX is true allow special effects if there are any defined. By saying allowfX=false, no special effects are used, even if some are defined.

184 \define@choicekey+{fitr}{allowFX}[\val\nr]{true,false}[true]%

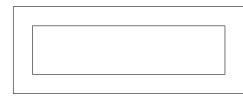
185 {\edef\fitr@allowFX{\val}}{}

186 \newcommand{\allowFXDefault}{true}

187 \let\fitr@allowFX\@empty

The main command. \jdRect (optionally) jumps to and/or sets a destination of a FitRectangle.\frac{1}{2} This command is multi-functional. This is the default behavior (button=true). The command creates two rectangular regions (which can be viewed using the preview and viewMagWin).

¹I had some problems naming this command.



The inner-box represents the bounding rectangle of the push button (or link) that is created when button=true (or link=jmp|restore). The outer rectangle is the view window, the window that is magnified to (the dimensions of this rectangle set by the four FitR parameters. Normally, the inner and outer rectangles are the same, unless the author specifies values of adddestw and adddesth. The picture above is that can be produced by setting

```
width=2in,height=.5in,adddestw=.2in,adddesth=.2in
```

The inner rectangle may be seen by using the option preview, the outer once can be see by using the option viewMagWin.

Now if button=false, \jdRect only sets the destination; it does not create the inner push button. A separate link or push button needs to be created to jump to the view window.

Syntax for \jdRect has two forms:

The above version is used to overlay a region with a button and view window. No content is specified, but is defined by specifying the width and height; it can be positioned using shift and lift.

There is a *-version as well:

The second parameter *content* is required when the * is present. This version is meant to enclose $\langle content \rangle$ within the button and view window. width and height are ignored, but shift and lift are obeyed (though you may shift or lift the button/view window away from the content).

```
188 \newcommand{\jdRect}{\begingroup
```

189 \@ifstar{\let\fitr@istar\ef@One\set@rectjd}%

90 {\let\fitr@istar\ef@Zero\set@rectjd}}

For links, we use the link color of hyperref. The definition of \ef@colorthislink is in eforms.

```
191 \@eqlinktxtcolor{\@linkcolor}
```

192 \newcommand{\fitr@bcode}{\ef@colorthislink}%

Second stage in the expansion of \jdRect.

193 \newcommand{\set@rectjd}[1][]{\def\fitr@jdrect@argi{#1}%

We process the key-values before gathering the \collectbox so we can insert the link color, if this is a link!

```
194 \edef\temp@exp{\noexpand\setkeys{fitr}{#1}}\temp@exp
```

95 \ifx\fitr@@link\@empty\let\fitr@bcode\relax\fi

227

```
If this is \jdRect*, we collect the second argument using \collectbox, and move
on to \set@@rectjd.
196
     \ifx\fitr@istar\ef@One
197
       \def\fitr@next{\collectbox[\fitr@bcode]{\set@@rectjd}}\else
Otherwise, we just move on to \set@@rectid.
198
       \def\fitr@next{\set@@rectjd}\fi\fitr@next}
The third and final stage in the expansion of \jdRect
199 \def\set@@rectjd{%
     \stepcounter{magCnt}\def\fitr@setBL{0pt}%
200
     \ifx\fitr@@dest\@empty
201
       \def\fitr@namedDest{fitrDestn\the\value{magCnt}}\else
202
       \edef\fitr@namedDest{\fitr@@dest}\fi
203
     \ifx\fitr@istar\ef@One\def\fitr@@refPt{b}\fi
204
If the user specifies the link option, we cancel the button option.
     \ifx\fitr@@link\@empty\else\KV@fitr@buttonfalse\fi
If content is explicitly passed in the optional second parameter we put it in a box
and take its measurements. We'll insert the content instead of an empty filled
 \parbox.
     \ifx\fitr@istar\ef@One
206
207
       \edef\fitr@@width{\the\wd\collectedbox}%
208
       \edef\fitr@setBL{\the\dp\collectedbox}%
       \setlength{\fitr@length}{\ht\collectedbox+\dp\collectedbox}%
209
       \edef\fitr@@height{\the\fitr@length}%
210
211
Now, we fit the content into \fitr@bbox using \get@fitr@dimen.
     \ifx\fitr@@link\@empty
212
       \ifKV@fitr@button\let\fitr@FLB@ction\fitr@OverlayJmpBtn\else
213
 \Bbox is defined in eforms.
214
       \@ifundefined{ef@Bbox}{\let\fitr@FLB@ction\Bbox}%
         {\let\fitr@FLB@ction\ef@Bbox}\fi
215
216
       \else\def\fitr@FLB@ction{\hfill\vfill\@gobbletwo}\fi
And overlay the button/link/box
     \get@fitr@dimen{%
217
       \parbox[\fitr@@refPt][\fitr@@height]{\fitr@@width}%
218
       {\kernOpt\fitr@FLB@ction{\fitr@@width}{\fitr@@height}\kernOpt}}%
219
Now place the button/box according to the user's options.
220
     \makebox[0pt][1]{\hspace*{\fitr@@shift}\smash{%
221
       \raisebox{\fitr@@lift-\fitr@setBL}{%
222
       \ifKV@fitr@nodest\else\setFitRDest{\fitr@@adddestw}%
         {\fitr@@adddesth}{\fitr@namedDest}\fi
223
       \unhbox\fitr@bbox\ifKV@fitr@nodest\else\fitr@urxury@fixup
224
         {\fitr@@adddesth}{\fitr@mamedDest}\fi
225
226
     }}}%
     \setlength{\fboxsep}{0pt}%
```

We attempt to represent the bounding box of the view window. We display the bounding view window only if the nodest option was not taken, and the \ifviewMagWin is true.

```
\ifKV@fitr@nodest\else\ifviewMagWin
228
       \fitr@length=\fitr@@adddestw bp
229
230
       \fitr@length=2\fitr@length
       \addtolength{\fitr@length}{\fitr@@width}%
231
       \edef\fitr@@width{\the\fitr@length}%
232
233
       \fitr@length=\fitr@@adddesth bp
       \fitr@length=2\fitr@length
234
       \addtolength{\fitr@length}{\fitr@@height}%
235
236
       \edef\fitr@@height{\the\fitr@length}%
237
       \setlength\fitr@length{-\fitr@@adddestw bp}%
       \addtolength\fitr@length{\fitr@@shift}%
238
239
       \makebox[0pt][1]{\hspace*{\fitr@length}\smash{%
         \setlength\fitr@length{\fitr@@lift-\fitr@setBL}%
240
         \if\fitr@@refPt b%
241
           \addtolength\fitr@length{-\fitr@@adddesth bp}\else
242
243
         \if\fitr@@refPt t%
           \addtolength\fitr@length{\fitr@@adddesth bp}%
244
245
         \fi\fi
246
         \raisebox{\fitr@length}%
         {\fbox{\parbox[\fitr@@refPt][\fitr@@height]%
247
         {\fitr@@width}{\kernOpt\hfill\vfill\kernOpt}}}}%
248
249
       }%
250
     \fi\fi
```

If the user passes content through the optional first parameter, we typeset it to the right of the material above (which did not change the position of the current point.

```
251
     \if\fitr@@link\ef@Zero
252
       \let\fitr@FLB@ction\fitr@OverlayJmpLnk\else
253
     \if\fitr@@link\ef@One
       \let\fitr@FLB@ction\fitr@OverlayRestoreLnk\else
254
     \let\fitr@FLB@ction\relax\let\fitr@bcode\relax\fi\fi
255
     \ifx\fitr@istar\ef@One
256
       \gdef\fitr@next{\fitr@FLB@ction{\BOXCONTENT}}\else
257
     \let\fitr@next\relax\fi\fitr@next
258
     \endgroup}
259
 The code for the FitR destination.
```

```
260 \def\setFitRDest#1#2#3{%
261 \if@fitr@dvipdfm
262 \FitRbboxB{#1}{#2}{#3}%
263 \else
264 \ifpdf
265 \FitRbboxB{#1}{#2}%
```

We shift the rectangle #1bp to the left without changing the current point. Then use the pdftex primitive \pdfdest.

```
266 \makebox[Opt][1]{\hspace*{-#1bp}%
267 \pdfdest name {#3} fitr
268 width \fitr@pdftex@view@width\space
269 height \fitr@pdftex@view@height\space
270 depth \fitr@pdftex@view@depth\space
271 }%
```

The **PostScript** code for the **FitR** destination, for the case of dvips and dvipsone.

```
272 \else\literalps@out{%

273 [ /Dest/#3/View [ /FitR \FitRbboxB{#1}{#2} ]/DEST pdfmark}%

274 \fi\fi

275 }
```

\overlayPresets

299

\Named{GoBack}\fi}]{#1}}

Form fields. The eforms package allows for what is called presets. We gather some useful options in one group, and pass them all together. \overlayPresets sets some defaults for the push button that surrounding the content. The \restoreOverlayPresets restores the default definition for \overlayPresets.

\restoreOverlayPresets

The \fitr@OverlayJmpBtn is a \pushButton that is used internally. It calls document JavaScript pbJmpBtnAction() for a mouse up action.

```
279 \def\fitr@OverlayJmpBtn#1#2{%
     \ifx\fitr@allowFX\@empty
       \def\allowFXcode{bAllowFX=\allowFXDefault}\else
281
       \def\allowFXcode{bAllowFX=\fitr@allowFX}\fi
282
     \pushButton[\presets{\overlayPresets}\F{-\FPrint}\autoCenter{n}
283
     \A{\ifFRusedljs\JS{\allowFXcode;\r
284
       pbJmpBtnAction(event,"\fitr@namedDest","\fitr@namedDest");}\else
285
       \GoToD(\fitr@namedDest)\fi}]{\fitr@namedDest}{#1}{#2}}
287 \ifHy@colorlinks\def\pbJmpLnkPresets{}\else
     \def\pbJmpLnkPresets{\Color{\@linkcolor}\W{1}}
288
289 \fi
290 \def\fitr@OverlayJmpLnk#1{\setLink[%
     \presets{\pbJmpLnkPresets}
291
     \A{\ifFRusedljs\JS{% dps
292
293
       pbJmpLnkAction("\fitr@namedDest");}\else
294
       \GoToD(\fitr@namedDest)\fi}]{#1}}
295 \def\fitr@OverlayRestoreLnk#1{\setLink[%
     \presets{\pbJmpLnkPresets}
296
     A{\left| ifFRusedljs\right| JS{% dps} }
297
       pbRestoreLnkAction("\fitr@namedDest");}\else
298
```

3 Document JavaScript

```
The document JavaScript for fitr.
                     300 \ifFRusedljs
                     301 \begin{insDLJS*}[_fitrLoaded]{fitr}
                     302 \begin{newsegment}{JS for the fitr Package}
                     303 var _fitrLoaded=true;
                     304 var restoreViewState;
                     305 var savedRestore=false;
                     306 var oSavedRestore=new Object();
                     307 var bAllowFX=true;
    pbJmpBtnAction(event,fname,dname) This is the function that the transparent overlay button calls
                      when the user clicks on it.
                     308 function pbJmpBtnAction(event,fname,dname) {
                     309
                             if (event.shift) shiftRestoreView(fname,dname);
                     310
                             else jumpToDest(fname,dname);
                     311 }
   shiftRestoreView(fname, dname) The function to call to restore a previously save view.
                     312 function shiftRestoreView(fname,dname) {
                            var bRestore=restoreView(fname,dname);
                      Try built-in store view special effects
                             if (bAllowFX) try {pbRestoreHook(event,bRestore);}catch(e){}
pbRestoreHookCustom(fname,dname) A custom function to perform special effects on restore.
                     315
                             if (bAllowFX) try {pbRestoreHookCustom(event,bRestore);}catch(e){}
                     316 }
         jumpToDest(fname,dname) When the user clicks on the overlay button, pbJmpBtnAction() calls
                      jumpToDest. The function stores the current state before jumping.
                     317 function jumpToDest(fname,dname) {
                             if ( typeof oSavedRestore[fname] == "undefined")
                     318
                     319
                                 oSavedRestore[fname] = [false, {}];
                     320
                      If oSavedRestore[fname] [0] is true, that means we have zoomed into the ex-
                      pression, but have not zoomed out again. We'll go ahead and zoom out by calling
                      shiftRestoreView(), which does just that.
                     321
                                 if (oSavedRestore[fname][0]) {
                                     shiftRestoreView(fname,dname)
                     322
```

When restoring, we return from this call; we don't want to jump to the destination, which is executed later.

```
Jump to the named destination.
                            this.gotoNamedDest(dname);
                      Try the built-in zoom-in special effects.
                            if (bAllowFX) try { pbJmpHook(event); }catch(e){}
   pbJmpHookCustom(event) is a custom function that is executed as part of the zoom-in action.
                            if (bAllowFX) try { pbJmpHookCustom(event); }catch(e){}
                     332
                     333 }
       restoreView(fname, dname) When the user shift-clicks on the overlay button, pbJmpBtnAction()
                      calls restoreView. This function restores the view before the user magnified the
                      view.
                     334 function restoreView(fname,dname) {
                            if ( typeof oSavedRestore[fname] == "undefined")
                     335
                     336
                                 oSavedRestore[fname]=[false,{}];
                     337
                            if ( oSavedRestore[fname][0] ) {
                     338
                                 this.viewState=oSavedRestore[fname][1];
                                 oSavedRestore[fname][0]=false;
                     339
                                 return true;
                     340
                            }
                     341
                     342
                            else return false;
                     343 }
    pbJmpLnkAction(\langle dname \rangle) is used when the link=jmp key-value pair is specified. We jump to the
                      named destination (dname). A link does not have a name, so we don't use an array
                      of stored view state as we do for push buttons. We use a single variable for all
                     links.
                     344 function pbJmpLnkAction(dname) {
                            if (!savedRestore){
                     345
                                 restoreViewState=this.viewState;
                     346
                     347
                                 savedRestore=true;
                            }
                     348
                            this.gotoNamedDest(dname);
                     349
                     350 }
pbRestoreLnkAction(\langle dname \rangle) is used when the link=restore key-value pair is specified. We restore
                      the previously saved view state.
                     351 function pbRestoreLnkAction(dname) {
                     352
                            if (savedRestore){
                                 this.viewState=restoreViewState;
                     353
                     354
                                 savedRestore=false;
                     355
                            }
                     356 }
                     357 \end{newsegment}
                     358 \ifFRblinkonjmp
                      This script hooks into the DLJS of fitr to blink the field border when the user
                      clicks in it. Length of the blink is 1250 milli-seconds.
                     359 \begin{newsegment}{Blink border after jump}
```

pbJmpHook(event) blinks the border of an pb push button just after the jmp to focus in on a rectangular region.

```
360 function pbJmpHook(event) {
361
       toggleBC.field=event.target;
362
       toggleBC.startColor=event.target.strokeColor;
       toggleBC.altColor=%
363
364 \; ({\tt color.equal(toggleBC.startColor,color.transparent)})?\%
365 color.red:color.transparent;
       oSIJ=app.setInterval("toggleBC();",250);
       oTOJ=app.setTimeOut("app.clearInterval(oSIJ); resetBC();",1250);
367
368 }
369 if (typeof toggleBC != "function" ) {
       function toggleBC() {
370
           var oField=toggleBC.field;
371
372
           oField.strokeColor=%
373 (color.equal(oField.strokeColor,toggleBC.startColor))?%
374 toggleBC.altColor:toggleBC.startColor;
375
       }
       function resetBC() {
376
377
           toggleBC.field.strokeColor=toggleBC.startColor;
378
       }
379 }
380 \end{newsegment}
381 \fi % \ifFRblinkonjmp
382 \ifFRblinkonrestore
```

This script hooks into the DLJS of fitr to blink the field border when the user clicks in it. Length of the blink is 1250 milli-seconds.

383 \begin{newsegment}{Blink border on restore}

pbRestoreHook(event, bRestore) Supplies special effects on restore: event is the event object that called this function; bRestore is a Boolean switch, it is true when we are restoring, and false when we are not.

```
384 function pbRestoreHook(event,bRestore) {
385
       if (!bRestore) return;
386
       toggleBC.field=event.target;
387
       toggleBC.startColor=event.target.strokeColor;
       toggleBC.altColor=%
388
389 (color.equal(toggleBC.startColor,color.transparent))?%
390 color.red:color.transparent;
       oSIR=app.setInterval("toggleBC();",250);
391
       oTOR=app.setTimeOut("app.clearInterval(oSIR); resetBC();",1250);
392
393 }
394 if (typeof toggleBC != "function" ) {
395
       function toggleBC() {
           var oField=toggleBC.field;
396
397
           oField.strokeColor=%
398 (color.equal(oField.strokeColor,toggleBC.startColor))?%
399 toggleBC.altColor:toggleBC.startColor;
400
```

```
401 function resetBC() {
402 toggleBC.field.strokeColor=toggleBC.startColor;
403 }
404 }
405 \end{newsegment}
406 \fi % \ifFRblinkonrestore
407 \end{insDLJS*}
408 \fi % \ifFRusedljs
409 \/package\
```

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 \@eqlinktxtcolor 191 \@fitr@dvipdfmfalse 25 \@fitr@dvipdfmtrue 27, 32, 37 \@linkcolor 191, 288 \@pdfm@mark 132, 137, 138	dvipdfmx (option) 2 dvips (option) 2 dvipsone (option) 2 E L \ef@Bbox 215 \ef@colorthislink 192
!blink (option) 4 !blinkonjmp (option) 3 !blinkonrestore (option) 4 !preview (option) 2, 3 !viewMagWin (option) 2, 3	\ef@Cone
Α	${f F}$
\A	\F 283 \fbox 247 \fboxsep 227
\afterassignment	\fitr@@adddesth 173, 174, 223, 225, 233, 242, 244 \fitr@@adddestw 170, 171, 222, 225, 229, 237 \fitr@@dest 182, 183, 201, 203
\allowFXDefault	\fitr@@height 89, 164, 165, 210, 218, 219, 235, 236, 247 \fitr@@lift
В	\fitr@QrefPt 167, 168, 204, 218, 241, 243, 247
\Bbox	\fitr@@shift
\BG	\fitr@allowFX
blinkonjmp (option) 3	\fitr@bcode 19, 61 66, 58, 115, 122, 123, 221 \fitr@bcode
blinkonrestore (option) 4 \BOXCONTENT 257	\fitr@defaultbp 141, 170, 173
button (key) 7	\fitr@depth
C \collectbox	\fitr@driver 8-11, 13-15, 17-19, 21-23, 26, 28, 29, 31, 33, 34, 36, 38, 39, 92, 104
\collectedbox	\fitr@dvipdfm@driver 30
\Color	\fitr@dvipdfmx@driver 35
	\fitr@dvips@driver 20, 92
D	\fitr@dvipsone@driver
\text{DeclareOptionX} \tag{9}, \tag{13, 17, 21, 26, 31, 36, 41-44, 50, 52, 53, 55-58}	\fitr@FLB@ction 213-216, 219, 252, 254, 255, 257 \fitr@height 74, 82, 101, 113
\define@boolkey 175, 180	\fitr@height@l 83, 135
\define@choicekey 166, 177, 184	\fitr@istar 189, 190, 196, 204, 206, 256
dest (key) 8	\fitr@jdrect@argi 193
\dimen@ 142, 147, 148	\fitr@length 77,
dvipdfm (option)	88, 89, 117–126, 209, 210, 229–240, 242, 244, 246

\fitr@namedDest	\JS 284, 292, 297
\dots 202, 203, 223, 225, 285, 286, 293, 294, 298	JS functions:
\fitr@next 197, 198, 257, 258	jumpToDest
\fitr@OverlayJmpBtn 213, 279	pbJmpHook
\fitr@OverlayJmpLnk 252, 290	pbJmpHookCustom
\fitr@OverlayRestoreLnk 254, 295	pbJmpLnkAction
\fitr@pdftex@driver 12, 16	pbRestoreHook
\fitr@pdftex@view@depth 126, 270	pbRestoreHookCustom
\fitr@pdftex@view@height 123, 269	pbRestoreLnkAction
\fitr@pdftex@view@width 120, 268	restoreView
\fitr@setBL 200, 208, 221, 240	jumpToDest (JS function)
\fitr@temp@length 151, 152, 155, 156, 159, 160, 163, 164	Jamp 102000 (00 Talleston)
\fitr@urxury@fixup 91, 134, 224	K
\fitr@width	\kern 219, 248
\fitr@width@l 85	keys:
\fitr@xetex@driver 40	adddesth 7
\FitRbboxB 5, 95, 107, 116, 129, 262, 265, 273	adddestw 7
\FPrint	allowFX 8
\FRblinkonjmpfalse 51, 53, 58	button 7
\FRblinkonjmptrue 52, 57	dest 8
\FRblinkonrestorefalse 54, 56, 58	height 7
\FRblinkonrestoretrue 55, 57	lift 7
\FRusedljsfalse 50	link 8
\FRusedljstrue 49	nodest 8
	ref 7
${f G}$	shift 7
\get@fitr@dimen <u>80</u> , 217	width 7
gonative (option) 3	\KV@fitr@buttonfalse 205
\GoToD 286, 294	\KV@fitr@buttontrue 176
Н	\KV@fitr@nodestfalse 181
\H 276, 278	
\headerps@out	${f L}$
height (key)	lift (key) 7
noight (hoy) ,	link (key) 8
I	\literalps@out 272
\if@fitr@dvipdfm	luatex (option) 2
\ifFRblinkonjmp	
\ifFRblinkonrestore 54, 382, 406	${f M}$
\ifFRusedljs 49, 284, 292, 297, 300, 408	\makeXasPDOff
\ifHy@colorlinks 287	
\ifKV@fitr@button 213	N
\ifKV@fitr@nodest 222, 224, 228	\Named 299
\ifluatex 60, 79	\newbox 73
$\verb \ \ \ \ \ \ \ \ \ \ \ \ \$	nodest (key) 8
$\verb \fifpreview 6$	0
$\verb \fiv if view Mag Win$	O
\ifxetex 62, 78	options:
J	!blink 4 !blinkonjmp 3
· ·	!blinkonrestore 4
\jdRect 8, 188	: DITHKUH eStule 4

!preview 2, 3 !viewMagWin 2, 3	\pushButton
blink 4	R
blinkonjmp 3	\r
blinkonrestore 4	\raisebox 130, 135, 221, 246
dvipdfm 2	ref (key)
dvipdfmx	RequirePackage 2-5, 67-71
dvips 2	\restoreOverlayPresets 12, 277
dvipsone	restoreView (JS function)
gonative 3	${f s}$
luatex	-
pbJmpBtnAction	\S
pdftex 2	
preview	\set@rectjd 189, 190, 193
shiftRestoreView	\setbox
viewMagWin	\setFitRDest
xetex	\setLink
\overlayPresets 12, 276, 278, 283	shift (key)
P	shiftRestoreView (option)
\PassOptionsToPackage 10,	\smash
11, 14, 15, 18, 19, 22, 23, 28, 29, 33, 34, 38, 39	\stepCounter
pbJmpBtnAction (option)	\Stripept140
pbJmpHook (JS function)	Т
pbJmpHookCustom (JS function)	\temp@exp 194
pbJmpLnkAction (JS function) 14	(vempeexp
\pbJmpLnkPresets 287, 288, 291, 296	U
pbRestoreHook (JS function)	\unhbox
pbRestoreHookCustom (JS function)	,
pbRestoreLnkAction (JS function)	\mathbf{V}
\pdfdest 79, 267	viewMagWin (option) 2, 3
\pdfextension 79	\viewMagWinfalse 7, 44, 48
pdftex (option) 2	\viewMagWinOff 3, 48
\presets 283, 291, 296	\viewMagWinOn 3, 47
preview (option)	\viewMagWintrue 43, 47
$\verb \previewfalse 6, 42, 46 $	·
$\verb \previewOff 3, 46$	\mathbf{W}
$\verb \previewOn 3, 45 $	\W 288
\previewtrue 41, 45	width (key) 7
\ProcessOptionsX	
\protected 79	\mathbf{X}
\providecommand 45-48	xetex (option) 2
5 Change History	
v1.3 (2020/07/03)	v1.3.1 (2020/07/06)
General: Add a test for xetex and luatex 4	General: Added \restoreOverlayPresets 12
	·
Added ifluatex package	Added blinkonjmp and blinkonjmp options 3

Added blinkonrestore and blinkonrestore	preview and viewMagWin options	3
options	v1.3.2 (2020/07/09)	
Added gonative option	General: Bad upload for v1.3.1, corrected	
Added commands to locally turn on the	problem	2