

GraphicxSP: Re-using EPS files

D. P. Story

Email: `dpsstory@acrotex.net & storyd@owc.edu`

processed December 1, 2019

Contents

1	Introduction	1
2	Package Options	1
3	Package Requirements	2
4	PostScript and Driver Dependent Definitions	2
5	Messing with <code>eso-pic</code>	4
6	Useful Supporting Commands	6
7	The Main Section	6
8	Messing with <code>graphicx</code>	13
9	Index	20
10	Change History	23

`1 <*package>`

1 Introduction

GraphicxSP is a patch into the `graphicx` package so that users of `dvips` and `dvipsone`, using distiller, can insert and re-use `.eps` figures.

2 Package Options

`preview` This package recognizes three options: driver names `dvips` (the default), `dvipsone` (old YandY TeX) and a `preview`. `dvipsone`, using distiller, can insert and re-use `.eps` figures.
`dvipsone`
`dvips`
`showembeds`
`!showembeds`

```

2 \@ifundefined{ifpreview}{\newif\ifpreview\previewfalse}{}
(2017/03/12) Added two convenience commands.
3 \providecommand{\previewOn}{\previewtrue}
4 \providecommand{\previewOff}{\previewfalse}
5 \DeclareOption{preview}{\previewtrue}
6 \DeclareOption{dvipsone}{\def\gxsp@drivernum{0}}
7 \DeclareOption{dvips}{\def\gxsp@drivernum{1}}
8 \DeclareOption{showembeds}{\let\gxsp@showembeds=0}
9 \DeclareOption{!showembeds}{\let\gxsp@showembeds=1}
10 \let\gxsp@showembeds=1
11 \def\gxsp@drivernum{1}

draft The draft mode passes draft on to graphicx. The images appear as rectangles,
final with the name of the image. The shownonnames option removes the name inside
shownonnames the rectangle.
!shownonnames 12 \DeclareOption{draft}{\spxGin@drafttrue
13     \PassOptionsToPackage{draft}{graphicx}}
14 \DeclareOption{!draft}{}
15 \DeclareOption{final}{\spxGin@draftfalse
16     \PassOptionsToPackage{final}{graphicx}}
17 \DeclareOption{shownonnames}{\@spx@shownameindraftfalse}
18 \DeclareOption{!shownonnames}{\@spx@shownameindrafttrue}
19 \newif\if@spx@shownameindraft \@spx@shownameindrafttrue
20 \newif\ifspxGin@draft \spxGin@draftfalse

21 \InputIfFileExists{graphics.cfg}{}{%
22 \ProcessOptions
23 \@ifundefined{eq@driver@name}{}{%
As a point of personal convenience, if \eq@driver@name, which is defined in web,
and the name is dvipsone, we'll override the default of dvips.
24 \def\DVIPSONE{dvipsone}\ifx\eq@driver@name\DVIPSONE
25 \def\gxsp@drivernum{0}\fi}

```

3 Package Requirements

We minimally require the `graphicx` package, which we patch into, and the `eso-pic`,
 package, which, in turn, requires the `everyshi` package.

```

26 \RequirePackage{graphicx}
27 \ifspxGin@draft\Gin@drafttrue\fi
28 \RequirePackage{eso-pic}
29 \RequirePackage{verbatim}

```

4 PostScript and Driver Dependent Definitions

`Hyperref` is not required, but if present, we'll use its code, otherwise, we use code
 from `hyperref` to hide in-line images from GhostScript's view.

```

30 \def\grcxsp@hideEPS{\AtBeginDvi{\special{!%
31 /product where{%
32 pop
33 product(Distiller)search{%
34     pop pop pop
35     userdict
36     /?pdfmark /exec load put%
37 }{%
38 pop
39 userdict
40 begin
41 /?pdfmark /pop load def
42 end
43 }ifelse%
44 }if%
45 }}}
46 \c@ifpackage{hyperref}{\let\grcxsp@hideEPS\relax}{\grcxsp@hideEPS}

We use either dvips or dvipsone as the driver, in both cases the following is the
special we shall use.

47 \def\gxsp@psMrk{[%]
48   \space}
49 \def\gxsp@literalps@out#1{\special{ps:#1}}

```

The following are driver dependent definitions. We begin with dvips.

Dvips driver.

```
50 \ifnum\gxsp@drivernum=1\relax
```

When the driver is dvips, we define some postscript procedures to making conversions between T_EX and PDF.

```

51 \special{!userdict begin
52   /TeXtoPDF {65536 div DVImag mul} def      % sp to pts
53   /PDFtoDvips {72.27 div Resolution mul} def  % points to dots
54   /PDFtoVDvips {72.27 div VResolution mul} def % points to dots
55   /DvipstoPDF {72.27 mul Resolution div} def  % dots to points
56   /HTeXtoDvips {TeXtoPDF PDFtoDvips} def      % sp to dots
57   /VTeXtoDvips {TeXtoPDF PDFtoVDvips} def end} % sp to dots

```

The `cstr` is used to calculate the lower left corner of the bounding box of an EPS file for dvips.

```

58 \special{!userdict begin /cstr {currentpoint translate
59   1 PDFtoDvips DVImag mul -1 PDFtoDvips DVImag mul scale}def end}
60 \def\gxsp@setPSCoordinate{cstr }
61 \def\b@grxsp@Literal{userdict begin}
62 \def\@e@grxsp@Literal{end}
63 \else

```

Dvipsone driver. The following is code special to the dvipsone driver.

The `undsc1x` is used to calculate the lower left corner of the the bounding box of an EPS file for `dvipsone`.

```
64 \def\gxsp@setPSCoor{undsc1x }
65 \let\b@grxsp@Literal\empty
66 \let\@grxsp@Literal\empty
67 \fi
```

These are procedures that support the dynamic naming of `/_objdef`. Distiller crashes if any symbolic reference name is not unique. So we must protect distiller. `grcxspObjDef` takes a single argument on the operand stack

```
(<name>) grcxspObjDef
```

and leaves at the top of the stack `(<name>)graphicxspCnt-currentpage)`

```
68 \special{!\b@grxsp@Literal
69   /currentpage 0 def
70   /graphicxspCnt 0 def
71   /graphicxspStr 10 string def
72   /graphicxspMergeStr {2 copy length exch length add string dup dup
73   4 3 roll 4 index length exch putinterval 3 1 roll exch
74   0 exch putinterval} def
75   /grcxspObjDef {
76     /graphicxspCnt graphicxspCnt 1 add def
77     currentpage graphicxspStr cvs graphicxspMergeStr
78     (-) graphicxspMergeStr
79     graphicxspCnt graphicxspStr cvs
80     graphicxspMergeStr
81 %    dup (Creating _objdef ) exch (\string\n) graphicxspMergeStr
82 %    graphicxspMergeStr print flush
83   } def \e@grxsp@Literal
84 }
```

5 Messing with `eso-pic`

One of the problems was to embed EPS files within a **BP/EP** operator pair. The solutions was to use `eso-pic`, place each graphic at the lower left corner of the page. We define a new “Hook” for `eso-pic` and attach it to `\@ShipoutPicture`.

```
85 \def\ESO@AeBip@Hook{}
86 \newcommand{\AddToEmbeddedEPSs}{\g@addto@macro\ESO@AeBip@Hook}
```

We redefine `\@ShipoutPicture` command of `eso-pic` so that the embedded figures are placed before `ESO@HookI` and `ESO@HookII`, for the case that someone wants to use placed pictures for a background, the file must be embedded before they can be inserted.

```
87 \@ifundefined{@ShipoutPicture}{%
```

The new version of `eso-pic` does not define `\@ShipoutPicture`, so we use some of the new code.

```
88   \ESO@isMEMOIR{%
```

```

89      \AtBeginShipout{%
90          \@tempdima=-\trimedge
91          \advance\@tempdima-\paperwidth
92          \advance\@tempdima\stockwidth
93          \if@twoside\ifodd\c@page\else
94              \advance\@tempdima2\trimedge
95              \advance\@tempdima\paperwidth
96              \advance\@tempdima-\stockwidth
97          \fi\fi
98          \tempdimb=\ESO@yoffsetI
99          \advance\tempdimb-\trimtop
100         \nointerlineskip
101         \AtBeginShipoutUpperLeft{%
102             \put(\LenToUnit{\@tempdima},\LenToUnit{\@tempdimb}){%
103                 \ESO@HookIII\ESO@HookI\ESO@HookII
104                 \global\let\ESO@HookII\empty
105             }%
106         }%
107     }%
108 }%
109     \AtBeginShipout{%
110         \nointerlineskip
111         \AtBeginShipoutUpperLeft{%
112             \put(0,\LenToUnit{\ESO@yoffsetI}){%
113                 \ESO@HookIII\ESO@AeBip@Hook\ESO@HookI\ESO@HookII% dps
114                 \global\let\ESO@HookII\empty
115                 \global\let\ESO@AeBip@Hook\empty% dps
116             }%
117         }%
118     }%
119 }
120 }{%

```

If \ShipoutPicture is defined, we use the old code.

```

121     \renewcommand{\ShipoutPicture}{%
122         \bgroup
123             \tempswafalse%
124             \ifx\ESO@HookI\empty\else\tempswatrue\fi%
125             \ifx\ESO@HookII\empty\else\tempswatrue\fi%
126             \ifx\ESO@HookIII\empty\else\tempswatrue\fi%
127             \ifx\ESO@AeBip@Hook\empty\else\tempswatrue\fi% dps(08/16/07)
128             \if@tempswa%
129                 \tempdima=1in\tempdimb=-\tempdima%
130                 \advance\tempdimb\ESO@yoffsetI%
131                 \ESO@isMEMOIR{%
132                     \advance\tempdima\trimedge%
133                     \advance\tempdima\paperwidth%
134                     \advance\tempdima-\stockwidth%
135                     \if@twoside\ifodd\c@page\else%
136                         \advance\tempdima-2\trimedge%

```

```

137          \advance\@tempdima-\paperwidth%
138          \advance\@tempdima\stockwidth%
139          \fi\fi%
140          \advance\@tempdimb\trimtop}%
141          \unitlength=1pt%
142          \global\setbox\@cclv\vbox{%
143              \vbox{\let\protect\relax
144                  \pictur@(0,0)(\strip@pt\@tempdima,\strip@pt\@tempdimb)%
145                  \ESO@HookIII\ESO@AeBip@Hook\ESO@HookI\ESO@HookII%dps
146                  \global\let\ESO@HookII\@empty%
147                  \global\let\ESO@AeBip@Hook\@empty%                      %dps
148                  \endpicture}%
149                  \nointerlineskip%
150                  \box\@cclv}%
151          \fi
152      \egroup
153  }
154 }

155 \AddToShipoutPicture{\special{ps: /currentpage \thepage\space def}}

```

6 Useful Supporting Commands

Some standard code that I use in AeB to wrote verbatim tex to a file.

```

156 \def\verbatimwrite{@bsphack
157     \let\do\@makeother\dospecials
158     \catcode`\^^M\active \catcode`\^^I=12
159     \def\verbatim@processline{%
160         \immediate\write\verbatim@out
161         {\the\verbatim@line}}%
162     \verbatim@start
163 }
164 \def\endverbatimwrite{@esphack}
165 \def\gxsp@IWV0{\immediate\write\verbatim@out}
166 \def\x@namedef#1{\expandafter\xdef\csname #1\endcsname}
167 \def\@namedef#1{\expandafter\edef\csname #1\endcsname}

```

Below is a counter to ensure each name is unique. It is used in \Ginclude@eps@SP.

```
168 \newcount\grxsp@cnt \grxsp@cnt=0
```

7 The Main Section

In this section we define two commands for the user, \embedEPS and \insertEPS, defined additional keys for the graphicx package, and consequently, hook into the \includegraphics command.

Some helper commands to save the dimensions of the pictures as they are embedded using \embedEPS.

```
169 \def\grcxsp@setPictureDimen#1#2#3#4#5{%
```

```

170      \x@namedef{#1Gin@llx}{#2}\x@namedef{#1Gin@lly}{#3}%
171      \x@namedef{#1Gin@urx}{#4}\x@namedef{#1Gin@ury}{#5}%
172      \x@namedef{#1BBox}{#2 #3 #4 #5}%
173      \begingroup

```

Calculate the width and height of the EPS. If the lower-left corner is not (0,0), results may not be predictable.

```

174      \tempdima=#4bp
175      \advance\tempdima-#2bp
176      \tempdima=.99626\tempdima
177      \x@namedef{#1widthOf}{\strip@pt\tempdima}%
178      \tempdima=#5bp
179      \advance\tempdima-#3bp
180      \tempdima=.99626\tempdima
181      \x@namedef{#1heightOf}{\strip@pt\tempdima}%
182      \endgroup
183 }

```

\heightOf More helper commands for calculating the height, width and path of an embedded file. These can be used by the user, that's you.

```

\llxOf 184 \def\heightOf#1{\csname#1heightOf\endcsname}
\llyOf 185 \def\widthOf#1{\csname#1widthOf\endcsname}
\urxOf 186 \def\bboxOf#1{\csname#1BBox\endcsname}
\uryOf 187 \def\llxOf#1{\csname#1Gin@llx\endcsname}
\csOf 188 \def\llyOf#1{\csname#1Gin@lly\endcsname}
189 \def\urxOf#1{\csname#1Gin@urx\endcsname}
190 \def\uryOf#1{\csname#1Gin@ury\endcsname}

```

Use `\csOf` to expand a name.

```
191 \let\csOf\nameuse
```

Other internal commands that save info.

```

192 \%def\grcxsp@pathOf#1{\csname#1path\endcsname}
193 \%def\grcxsp@importSF#1{\csname#1importScaleFactor\endcsname}

```

We redefine a command from `graphics`. When testing the `draft` option, we had some problems with an underscore `_` in the value of the `name` key, so we sanitize this character.

```

194 \%def\spx@sanitize{\catcode`\_=12\relax}
195 \%def\Gin@i{%
196 %  \ifnextchar[%]
197 %    {\spx@sanitize\Gin@ii}
198 %    {\Gin@bboxfalse\Ginclude@graphics}}

```

\embedEPS This is the command for embedding an EPS file in the document for use by the `SP` operator. The command takes three arguments, one of which is optional

[#1]: Recognizes two key-value pairs (1) `hiresbb`, this is the same key-value used by the `graphicx` package; (2) `transparencyGroup`, a new option for creating a transparency group. Using `transparencyGroup` without any value will

make the embedded graphic into a transparency group, with a value adds additional keys as documented in the *PDF Reference*.

#2: A symbolic name for the embedded graphic, this name is used by distiller.

#3: path to the EPS file (without extension).

I prefer the \embedEPS commands to appear in the preamble, but they can appear anywhere before the first appearance of \includegraphics or \insertEPS that reference the embedded file. I suppose this embedding could have been automatic at the first occurrence of \includegraphics or \insertEPS, but I didn't go that route.

```

199 \newcommand{\embedEPS}[3] []{%
200   \Gin@undefined{\#2\Gin@llx}{\Gin@lly}{%
201     \PackageError{graphicx}{%
202       The name, #2, on line \the\inputlineno\MessageBreak
203       is already defined. All embedded graphics\MessageBreak
204       must be assigned a unique name}%
205     {Give this embedded graphic a unique name.}%
206   }%
207   \begingroup
208     \let\Gin@transparencygroup\empty

```

We use the graphicx command \Gread@eps to verify that the graphic exists, and if so, get its bounding box parameters. We work only with .eps files so let's add the extension.

```

209   \let\input@path\Ginput@path
210   \filename@parse{\#3.eps}%
211   \Gin@getbase{.eps}%
212   \Gin@undefined{\Gin@base}{%
213     \PackageError{graphicx}{%
214       Graphics file #3 specified on \the\inputlineno\MessageBreak
215       was not found}%
216   }%
217   Verify the file exists, is an eps file,\MessageBreak
218   is on the latex search path, or is in the\MessageBreak
219   current directory.%%
220 }%
221 }%
222 }{%
223 \e@namedef{gxsp@\Gin@base}{\Gin@base}%
224 \Gread@eps{\gxsp@\Gin@base.eps}%

```

Now set the keys. We delayed the \setkeys because name=#2 would set the switch \if@Ginnamed to true, which has consequences on computing the \Gin@base when \Gin@setfile is executed.

```
225   \setkeys{Gin}{name=#2,#1}%

```

Once the file is found and the bounding box parameters are recorded by graphicx, we save these under the graphic's embedded symbolic name.

```

226      \grcxsp@setPictureDimen%
227          {\Gin@name}{\Gin@llx}{\Gin@lly}{\Gin@urx}{\Gin@ury}%
228 % \x@namedef{\Gin@name path}{#3}%

```

If an embedded graphic exceeds the boundaries of the paper size, the graphic is clipped off. What I am doing below is determining the largest scale factor, `\gxsp@embedSF`, needed to embed the file without exceeding the page boundaries.

```

229      \def\gxsp@embedSF{1}%
230      \tempdima=\Gin@urx bp
231      \advance\tempdima-\Gin@llx bp
232      \ifdim\tempdima>\paperwidth
233          \Gscale@div\gxsp@embedSF\paperwidth\tempdima
234          \tempdima=\Gin@ury bp
235          \advance\tempdima-\Gin@lly bp
236          \tempdima=\gxsp@embedSF\tempdima
237          \ifdim\tempdima>\paperheight
238              \edef\gxsp@embedSFSave{\gxsp@embedSF}%
239              \Gscale@div\gxsp@embedSF\paperheight\tempdima
240              \tempdima=\gxsp@embedSFSave\p@
241              \tempdima=\gxsp@embedSF\tempdima
242              \edef\gxsp@embedSF{\strip@pt\tempdima}%
243          \fi
244      \else
245          \tempdima=\Gin@ury bp
246          \advance\tempdima-\Gin@lly bp
247          \ifdim\tempdima>\paperheight
248              \Gscale@div\gxsp@embedSF\paperheight\tempdima
249          \fi
250      \fi

```

Now that we have `\gxsp@embedSF`, we add the current graphic to our collection of embedded files using `\AddToEmbeddedEPSs`, which is a variation on `\AddToShipoutPicture`, but uses `\ESO@eBip@Hook` for our private use. We expand some of the arguments before executing `\AddToEmbeddedEPSs`.

```

251      \edef\tempa{%
252      \noexpand\AddToEmbeddedEPSs{\noexpand\AtPageLowerLeft{%
253          \noexpand\scalebox{\gxsp@embedSF}{%
254              \noexpand\gxsp@embedEPS{\gxsp@Gin@base}{\Gin@name}}}}%
255          \noexpand\AtPageCenter{\noexpand\gcxsp@wrapEmbeddedFigure{%
256              \Gin@transparencygroup}{\Gin@transparency}}}\tempa
257      \endgroup
258 }
259 \onlypreamble{\embedEPS}

```

In a dvi previewer, the embedded graphics are visible on the first page. The `\grcxsp@coverEmbeds` puts a white color box over the graphics, L^AT_EX content and other graphics are placed over this white color box. The white color box can be removed with the `showembeds` option.

```

260 \def\grcxsp@coverEmbeds{%
261     \AddToEmbeddedEPSs{\AtPageLowerLeft{\colorbox{white}{%

```

```

262     \parbox[b]{\paperheight}{\paperwidth}{\hfill\vfill}}}}}
263 \if\gxsp@showembeds1%
264 \AtBeginDocument{\grcxsp@coverEmbeds}
265 \else
266 \let\grcxsp@coverEmbeds\relax
267 \fi

```

\gxsp@embedEPS The \gxsp@embedEPS command embeds the file, and is called by \embedEPS. It takes three options: (1) the value of `transparencyGroup`; (2) the EPS path; and (3) the symbolic name for the graphic.

The bounding box /BBox acts as a clipping path, if the graphic falls outside the box, it is clipped off. Since we don't know the size of the graphic in advance, and the value of the `%%BoundingBox` can be deceiving, set the of /BBox to an array with enormous dimensions, the default is `\grcxsp@maxDim = 5000`. This can be reset to larger value if you are embedding graphics of even more enormous dimensions.

```
268 \def\grcxsp@maxDim{5000}
```

Now, for the \gxsp@embedEPS command that embed the graphic between **BP** and **EP**.

```

269 \newcommand{\gxsp@embedEPS}[2]{%
270   \gxsp@literalps@out{\gsave \gxsp@setPSCoor
271   \gxsp@psMrk/BBox [-\grcxsp@maxDim\space-\grcxsp@maxDim\space
272   \grcxsp@maxDim\space\grcxsp@maxDim]\space/_objdef {Embedded:#2}
273   /BP pdfmark grestore}%
274   \message{<Embedding #1>}%

```

If we are using dvipsone, we can suppress the preview of the embedded file by not using the extension. Dviwindo will look for a tiff file, if not present, will not display a preview.

```

275   \includegraphics{#1}%
276   \gxsp@literalps@out{\gxsp@psMrk/EP pdfmark}%
277 }
```

We create a wrapper that shows the Embedded file under the original symbolic name. Here we introduce any transparency ordered up in the option list of \embedEPS

```

278 \def\gcxsp@wrapEmbeddedFigure#1#2#3{%
279   \def\Gin@transparencygroup{#1}\def\Gin@transparency{#3}%
280   \gxsp@literalps@out{\gsave \gxsp@setPSCoor
281   \ifGin@clip
282     [/BBox [\llx0f{#2}\space\lly0f{#2}\space
283     \urx0f{#2}\space\ury0f{#2}]
284   \else
285     [ /BBox [-\grcxsp@maxDim\space-\grcxsp@maxDim\space
286     \grcxsp@maxDim\space\grcxsp@maxDim]
287   \fi\space/_objdef {#2}
288   \ifx\Gin@transparencygroup\empty\else
289     \ifx\Gin@transparencygroup\Gin@exclamation
```

```

290         /Group << /S/Transparency >>%
291     \else
292         /Group << /S/Transparency \Gin@transparencygroup >>%
293     \fi
294 \fi\space
295 /BP pdfmark
296 \gxsp@psMrk{Embedded:#2} /SP pdfmark
297 \gxsp@psMrk/EP pdfmark
298 grestore}%
299 }

```

`createImage` The `createImage` environment can be used for two purposes:

1. Use it to take a file already embedded, manipulate it, and give it a symbolic name.
2. Use postscript graphic operators to create an image.

The images can be shown using `\includegraphics` or `\insertEPS`, or they can be referenced as an appearance of a form field.

We try something different. My usual approach for a verbatim environment is to write the contents to an auxiliary file and input that file back in. This approach precludes using the environment in another command. The text to this environment should be PostScript or PDF language statements, or TeX macros that expand to same. We'll absorb the contents in the environment as an argument #1 of the `\grxcsp@createImage` command. However, before we get to `\grxcsp@createImage` we must execute `\createImage`, the user's access to this code.

`\createImage` takes three arguments, the first one of which is optional

#1: Takes the key-values of `\includegraphics`, plus some of the `graphicxsp` key-values, such as `transparencyGroup`. The `name` key is ignored, and is declared in the third parameter.

#2: The bounding box for this image.

#3: The `name` to be attached to this image.

```

300 \def\ci@undef@msg#1{\PackageWarning{graphicxsp}{The command
301   '\expandafter\string\csname #1\endcsname'
302   is already defined\MessageBreak
303   choose a different name instead of\MessageBreak'#1')}
304 \newcommand{\sp@createImage}[3][]{%
305   \@ifundefined{#3}{}{\ci@undef@msg{#3}%
306   \x@namedef{#3}{#3}%
307   \ifundefined{#3Gin@llx}{}{%
308     \PackageError{graphicxsp}%
309     {The name, #3, on line \the\inputlineno\space\MessageBreak
310     is already defined. All embedded graphics\MessageBreak
311     must be assigned a unique name}

```

```

312     {Give this embedded graphic a unique name.}%
313   }%
314   \setkeys{Gin}{#1}\def\Gin@name{#3}%
315   \edef\@gtempa{#2 }%
316   \expandafter\Gread@parse@bb\@gtempa \\%
317   \begingroup\grxcsp@createImage
318 }
319 \let\postEP\empty
320 \long\def\grxcsp@createImage#1\end#2{%
321   \def\reserved@a{#2}\ifx\reserved@a\@currenvir
322     \end{#2}\else\@badend{#2}\fi
323   \edef\temp@transparencyGroup{%
324     \ifx\Gin@transparencygroup\empty\else
325       \ifx\Gin@transparencygroup\Gin@exclamation
326         /Group << /S/Transparency >>%
327       \else
328         /Group << /S/Transparency \Gin@transparencygroup >>%
329       \fi
330   }%
331   \grcxsp@setPictureDimen%
332   {\Gin@name}{\Gin@llx}{\Gin@lly}{\Gin@urx}{\Gin@ury}%

```

(2009/02/19) We use `\AddToEmbeddedEPSs` to embed EPS created by the `createImage` environment. This allows the EPS to be used on the first page, which has been a problem in the past.

```

333   \edef\@tempaf{%
334     \noexpand\AddToEmbeddedEPSs{\noexpand\AtPageLowerLeft{%
335       \noexpand\gxsp@literalps@out{gsave \gxsp@setPSCoor
336       \gxsp@psMrk/BBox
337       [\Gin@llx\space\Gin@lly\space\Gin@urx\space\Gin@ury]
338       /_objdef {\Gin@name} \temp@transparencyGroup\space/BP pdfmark
339       \ifx\Gin@transparency\empty\else
340         \gxsp@psMrk\Gin@transparency\space/SetTransparency pdfmark\fi
341       {#1} ?pdfmark
342       \gxsp@psMrk/EP pdfmark
343       grestore
344     }%
345   }}\@tempa
346   \endgroup
347 }
348 \let\createImage\sp@createImage
349 \let\endcreateImage\endsp@createImage
350 \onlypreamble{\createImage}

```

`\insertEPS` The idea was to use the `\includegraphics` command to show a graphic that has been earlier embedded. However, one of `\includegraphics` arguments is the path of the eps file. Once, the file is embedded, the path is not needed, so this package defines `\insertEPS`. This command takes two arguments: (1) The usual `\includegraphics` options, plus any other options defined in this package; (2) the symbolic name. Because the symbolic name is passed as the second argument,

it is not necessary to specify in the optional parameter list. The following two (should) be equivalent:

```
\embedEPS{myCoolSelfPic}
...
\begin{document}
...
\includegraphics[name=AdobeDon,width=1in]{myCoolSelfPic}
\insertEPS[width=1in]{AdobeDon}
...
351 \def\xsp@sanitize{\catcode`\_=12\relax}
352 \newcommand{\insertEPS}{\bgroup\xsp@sanitize
353   \Gifstar
354   {\Gin@cliptrue\let\gcxsp@star*\gcxsp@insertEPS}%
355   {\Gin@clipfalse\let\gcxsp@star\empty\gcxsp@insertEPS}%
356 \newcommand{\gcxsp@insertEPS}[2][]{%
357   \expandafter\includegraphics\gcxsp@star[name=#2,#1]{}\egroup}
```

8 Messing with `graphicx`

In this section, we add some options to the `graphicx` package. We define some additional keys that will be recognized by `\includegraphics`. We also redefine `\Gin@ii` and `\Gin@setfile`, which are `graphicx` commands to make things work for us.

- name** Use the `name` key-value pair only for graphics already embedded by `\embedEPS`. When this key is present, we `\let \Gininclude@eps` to `\Gininclude@eps@SP`. `\Gininclude@eps` is the usual way of handling EPS files, `\Gininclude@eps@SP` is how we are to handle files already embedded. Usage:

```
\includegraphics[name=AdobeDon,width=1in]{myCoolSelfPic}
358 \newif\if@Ginnamed\Ginnamedfalse
359 \define@key{Gin}{name}[]{\def\Gin@name{\#1}%
360   \Ginnamedtrue\let\Gininclude@eps\Gininclude@eps@SP}
361 \def\Gin@name{}
```

- transparencyGroup** This defines the `transparencyGroup` key which is used only recognized with `\embedEPS`. See the Transparency section of the pdfmark Reference and the chapter on Transparency in the PDF Reference. In particular, see PDF Ref Table 7.13. Usage:

```
\embedEPS[transparencyGroup]{myCoolSelfPic}
362 \define@key{Gin}{transparencyGroup}[]{\def\Gin@transparencygroup{\#1}}
363 \def\Gin@transparencygroup{}
```

- transparency** Enter any transparency postscript key-value pairs for this image. These are ignored unless the embedded file is a transparency group, and you distill with `<< /AllowTransparency true >>` setdistillerparams!. Usage:

```

\embedEPS[transparencyGroup]{myCoolSelfPic}
...
\begin{document}
...
\includegraphics[name=AdobeDon,width=1in,
    transparency={/ca .5 /BM/Normal}]{myCoolSelfPic}

or

\insertEPS[width=1in,transparency={/ca .5 /BM/Normal}]{AdobeDon}

364 \define@key{Gin}{transparency}[] {\def\Gin@transparency{\#1}}%
365 \def\Gin@transparency{}%
366 \define@key{Gin}{SMask}[] {\def\Gin@SMask{\#1}}%
367 \def\Gin@SMask{}%

```

- presp** We define two additional keys for creating special effects. The value of **presp** and **postsp** are postscript commands for manipulating the image. As the names suggest, **presp** is placed before the **SP** operator, and **postsp** is placed after. Example of usage is given in one of the demo files.

```

368 \define@key{Gin}{presp}{\def\Gin@presp{\#1}}
369 \def\Gin@presp{}
370 \define@key{Gin}{postsp}{\def\Gin@postsp{\#1}}
371 \def\Gin@postsp{}

```

The following key-value pairs are recognized by **\setSMask**, **\embedEPS** and **\includegraphics** and are used to set up a soft mask.

- SMask** The key **SMask** is used in the optional parameter list of **\insertEPS** and **\includegraphics**, when that graphic is to use a soft mask. The value of **SMask** is a key-value list, the keys are **subtype**, **group**, **bc** and **tr**. The default for **subtype** is **Luminosity**, the other value recognized is **Alpha**. If **subtype** is not listed, **Luminosity** is used for the **subtype**. The **group** key is required, and the latex compile will stop if it is not specified. The value of **group** is the name of a graphic to be used as a mask. This graphic must be a transparency group with the **CS** key specified. The other two keys, **bc** (component color) and **tr** (transfer function) to complete the supported keys. See Table 7.10, page 553, of the *PDF Reference*, Version 8, for detailed descriptions of these key-values.

```

372 \define@key{Gin}{SMask}[] {\def\GinSP@SMask{\#1}}
373 \def\GinSP@SMask{}
374 \define@key{GinSP}{subtype}[Luminosity]{\def\GinSP@subtype{\#1}}
375 \def\GinSP@subtype{Luminosity}
376 \define@key{GinSP}{group}[] {\def\GinSP@group{\#1}}
377 \def\GinSP@group{}
378 \define@key{GinSP}{bc}[] {\def\GinSP@bc{\#1}}
379 \def\GinSP@bc{}
380 \define@key{GinSP}{tr}[] {\def\GinSP@tr{\#1}}
381 \define@key{GinSP}{None}[None]{\def\GinSP@None{\#1}}
382 \def\GinSP@tr{}
383 \def\GinSP@None{}

```

```

384 \def\SMaskSP@Identity{Identity}

\setSMask Use \setSMask to set a soft mask. This command takes one required argument,
the name of the transparency group to be used as the source of alpha or color values
for deriving the mask. The optional parameter consists of key-value pairs for the
soft-mask dictionary, see table 7.10 of the PDF Reference, Version 8.

385 \def\sp@setSMask{%
386   \ifx\GinSP@SMask\SMaskSP@None\gxsp@psMrk/SMask/None
387     /SetTransparency pdfmark
388   \else
389     \gxsp@psMrk/SMask << /S/\GinSP@subtype\space
390     \ifx\GinSP@bc\empty\else/BC\GinSP@bc\space\fi
391     \ifx\GinSP@tr\empty\else\ifx\GinSP@tr\SMaskSP@Identity%
392       /TR/Identity\else/TR {\GinSP@tr}\space\fi\fi
393     /G {\GinSP@group} >> /SetTransparency pdfmark
394   \fi
395 }
396 \newcommand{\setSMask}[2][]{%
397   \setkeys{GinSP}{#1}\def\GinSP@group{#2}%
398   \special{ps: \sp@setSMask}%
399 }

```

We redefine `\Gin@setfile`. If the graphic is named, we salt things with the bounding box parameters.

```

400 \def\Gin@setfile#1#2#3{%
401   \ifx\\#2\\\Gread@false\fi
402   \ifGin@bbox\else
403     \ifGread@
404     \if@Ginnamed %dps (08/16/07)
405       \edef\Gin@llx{\csname\Gin@name\Gin@llx\endcsname}%
406       \edef\Gin@lly{\csname\Gin@name\Gin@lly\endcsname}%
407       \edef\Gin@urx{\csname\Gin@name\Gin@urx\endcsname}%
408       \edef\Gin@ury{\csname\Gin@name\Gin@ury\endcsname}%
409     \else
410       \csname Gread@\%
411         \expandafter\ifx\csname Gread@\#1\endcsname\relax
412           eps%
413         \else
414           #1%
415         \fi
416       \endcsname{\Gin@base#2}%
417     \fi
418     \else
419       \Gin@nosize{#3}%
420     \fi
421   \fi
422   \Gin@viewport@code
423   \Gin@nat@height\Gin@ury bp%
424   \advance\Gin@nat@height-\Gin@lly bp%

```

```

425  \Gin@nat@width\Gin@urx bp%
426  \advance\Gin@nat@width-\Gin@llx bp%
427  \Gin@req@sizes
428  \expandafter\ifx\csname Gininclude@\#1\endcsname\relax
429    \Gin@drafttrue
430    \expandafter\ifx\csname Gread@\#1\endcsname\relax
431      @latex@error{Can not include graphics of type: #1}@\ehc
432      \global\expandafter\let\csname Gread@\#1\endcsname@\empty
433    \fi
434  \fi
435  \leavevemode
436  \ifGin@draft
437    \hb@xt@\Gin@req@width{%
438      \vrule\hss
439      \vbox to \Gin@req@height{%
440        \hrule \Gin@req@width \Gin@req@width
441        \vss
442        \if@Ginnamed %dps (08/18/07)

```

If the `shownonames` option is taken, we do not show the name of the graphic.

```

443  \if@spx@shownameindraft
444    \rlap{ \ttfamily\Gin@name}\fi
445  \else
446    \edef\@tempa{#3}%
447    \rlap{ \ttfamily\expandafter\strip@prefix\meaning\@tempa}%
448    \fi
449    \vss
450    \hrule}%
451    \hss\vrule}%
452 \else
453 \if@Ginnamed\else % dps (08/16/07)
454   \addtofilelist{#3}%
455   \ProvidesFile{#3}[Graphic file (type #1)]%
456 \fi
457   \setbox\z@\hbox{\csname Gininclude@\#1\endcsname{#3}}%
458   \dp\z@\z@
459   \ht\z@\Gin@req@height
460   \wd\z@\Gin@req@width
461   \ifpreview{\setlength{\fboxsep}{0pt}\fbox{\box\z@}}\else\box\z@\fi%
462 \fi}
463 \def\Gin@getbase#1{%
464   \edef\Gin@tempa{%
465     \def\noexpand\@tempa####1#1\space{%
466       \def\noexpand\Gin@base{####1}}%

```

If the current graphic is named, then we don't need to read the bounding box again or to see if it exists again.

```

467  \if@Ginnamed
468    \edef\Gin@ext{#1}\edef\Gin@base{\Gin@name}%dps (08/18/07)
469  \else

```

If not named, we need to handle it in the usual way.

(2019/11/13) The latest L^AT_EX core defines `\IfFileExists@` which broke this package, we try a fix here.

```

470      \IfFileExists{\filename@area\filename@base#1}%
471      {\@ifundefined{IfFileExists@}{}}%
472      {\edef\@filef@und{\filename@area\filename@base#1 }%
473       \Gin@tempa\expandafter\@tempa\@filef@und
474       \edef\Gin@ext{\#1}{}%
475   \fi
476 }
```

`\Gin@computeSF` Based on scaling info provided by graphicx, we compute the scale factors we need.

```

477 \def\Gin@computeSF{\def\@tempa{!}%
478   \edef\gxsp@scaleFactor@x{\Gin@scalex}%
479   \edef\gxsp@scaleFactor@y{\Gin@scaley}%
480   \ifx\Gin@scaley\@tempa      % proportional height
481     \ifx\Gin@scalex\@tempa    % proportional width
482       \def\gxsp@scaleFactor@x{1}%
483       \def\gxsp@scaleFactor@y{1}%
484     \else                     % specified width
485       \edef\gxsp@scaleFactor@y{\Gin@scalex}%
486   \fi
487   \else                      % specified height
488     \ifx\Gin@scalex\@tempa  % proportional width
489       \edef\gxsp@scaleFactor@x{\Gin@scaley}%
490   \fi
491 \fi
492 }
```

`\Ginclude@eps@SP` This is the substitute for the usual way of handing an EPS file. Here we use the **SP** to show the embedded graphic.

```

493 \def\gxsp@setBBox{%
494   \ifGin@clip
495     [/BBox [\Gin@llx\space\Gin@lly\space\Gin@urx\space\Gin@ury]
496   \else
497     [/BBox [-\grcxsp@maxDim\space-\grcxsp@maxDim\space
498           \grcxsp@maxDim\space\grcxsp@maxDim]
499   \fi
500 }
501 \def\Ginclude@eps@SP#1{%
502 % \message{<#1>}%
503 \bgroup
```

See if the user has specified the **SMask** key, if yes, we'll check to see if a group name was specified. The group name is required. If no, we halt the compile job.

```

504   \ifx\GinSP@SMask\empty\else
505     \edef\sp@expand@temp{\noexpand\setkeys{GinSP}{\GinSP@SMask}}%
506     \sp@expand@temp
507     \ifx\SMaskSP@None\empty\ifx\GinSP@group\empty
```

```

508      \PackageError{graphicxsp}{The group key is required when you
509      specify a SMask.}{Specify a group name for the group key.}%
510      \fi\fi\fi
511      \Gin@computeSF
512      \ifGin@bbox
513      \gxsp@literalps@out{%
514          gsave \gxsp@setPSCoor

```

If SMask is specified, we call \sp@setSMask to set the graphics state parameters for a soft mask.

```

515          \ifx\GinSP@SMask@\empty\else\sp@setSMask\fi
516          \gxsp@setBBox\space

```

We push the basename (\Gin@name:bbox) and call grcxspObjDef. This procedure returns (\Gin@name:bboxgraphicxspCnt-currentpage). graphicxspretn then takes that result, and converts it to a name type. We then use it in /_objdef { // /graphicxspretn }, using immediate execution.

```

517          (\Gin@name:bbox@) grcxspObjDef
518          /graphicxspretn exch cvx cvn def
519          /_objdef { // /graphicxspretn } /BP pdfmark
520          \ifx\Gin@transparency@\empty\else
521          \gxsp@psMrk\Gin@transparency\space
522          /SetTransparency pdfmark\fi
523          \gxsp@psMrk{\Gin@name} /SP pdfmark
524          \gxsp@psMrk/EP pdfmark
525          \gxsp@scaleFactor@x\space\gxsp@scaleFactor@y\space scale
526          -\Gin@llx\space -\Gin@lly\space moveto
527          currentpoint translate
528          \Gin@presp
529          \gxsp@psMrk{ // /graphicxspretn } /SP pdfmark
530          \Gin@postsp
531          grestore
532      }%
533      \else
534      \gxsp@literalps@out{%
535          gsave \gxsp@setPSCoor

```

If SMask is specified, we call \sp@setSMask to set the graphics state parameters for a soft mask.

```

536          \ifx\GinSP@SMask@\empty\else\sp@setSMask\fi
537          \gxsp@setBBox\space
538          (\Gin@name:grxsp@) grcxspObjDef
539          /graphicxspretn exch cvx cvn def
540          /_objdef { // /graphicxspretn } /BP pdfmark
541          \ifx\Gin@transparency@\empty\else
542          \gxsp@psMrk\Gin@transparency\space
543          /SetTransparency pdfmark\fi
544          \gxsp@psMrk{\Gin@name} /SP pdfmark
545          \gxsp@psMrk/EP pdfmark
546          \gxsp@scaleFactor@x\space\gxsp@scaleFactor@y\space scale
547          \ifx\Gin@viewport@code\relax\else

```

```
548      -\Gin@llx\space-\Gin@lly\space moveto
549      currentpoint translate\fi
550      \Gin@presp
551      \gxsp@psMrk{ //graphicxspren } /SP pdfmark
552      \Gin@postsp
553      grestore
554      }%
555      \fi
556  \egroup}
557 </package>
```

9 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

\@Ginnamedfalse	358
\@Ginnamedtrue	360
\@ShipoutPicture	121
\@addtofilelist	454
\@badend	322
\@bsphack	156
\@cclv	142, 150
\@currenvir	321
\@ehc	431
\@esphack	164
\@filef@und	472, 473
\@gtempa	315, 316
\@ifnextchar	196
\@ifpackageloaded	46
\@ifstar	353
\@ifundefined	2, 23, 87, 200, 212, 305, 307, 471
\@latex@error	431
\@makeother	157
\@nameuse	191
\@onlypreamble	259, 350
\@spx@shownameindraftfalse	17
\@spx@shownameindrafttrue	18, 19
\@tempa	251, 256, 333, 345, 446, 447, 465, 473, 477, 480, 481, 488
\@tempdima	90–92, 94–96, 102, 129, 132–134, 136– 138, 144, 174–181, 230–237, 239–242, 245–248
\@tempdimb	98, 99, 102, 129, 130, 140, 144
\@tempswafalse	123
\@tempswatrue	124–127
\@width	440
\`	316, 401
\^	158
_	194, 351

A

\active	158
\AddToEmbeddedEPSS	86, 252, 261, 334
\AddToShipoutPicture	155
\AtBeginDocument	264
\AtBeginDvi	30
\AtBeginShipout	89, 109
\AtBeginShipoutUpperLeft	101, 111
\AtPageCenter	255
\AtPageLowerLeft	252, 261, 334

B

\b@grxsp@Literal	61, 65, 68
\bboxOf	186
\bc	<u>372</u>
\begingroup	173, 207, 317
\bgroup	122, 352, 503
\box	150, 461

C

\c@page	93, 135
\catcode	158, 194, 351
\ci@undef@msg	300, 305
\colorbox	261
\createImage	348, 350
\createImage (environment)	<u>300</u>
\csname	166, 167, 184–190, 192, 193, 301, 405–408, 410, 411, 428, 430, 432, 457
\cs0f	<u>184</u>

D

\DeclareOption	5–9, 12, 14, 15, 17, 18
\def	6, 7, 11, 24, 25, 30, 47, 49, 60–62, 64, 85, 156, 159, 164–167, 169, 184–190, 192–195, 229, 260, 268, 278, 279, 300, 314, 320, 321, 351, 359, 361–385, 397, 400, 463, 465, 466, 477, 482, 483, 493, 501
\define@key	359, 362, 364, 366, 368, 370, 372, 374, 376, 378, 380, 381
\do	157
\dospecials	157
\dp	458
\draft	<u>12</u>
\dvips	<u>2</u>
\DVIPSONE	24
\dvipson	<u>2</u>

E

\e@grxsp@Literal	62, 66, 83
\e@namedef	167, 223
\edef ..	167, 238, 242, 251, 315, 323, 333, 405–408, 446, 464, 468, 472, 474, 478, 479, 485, 489, 505
\egroup	152, 357, 556
\else	63, 93, 124–127, 135, 244, 265, 284, 288, 291, 322, 324, 327, 339, 388, 390–392,

\endcsname	166, 167, 184–190, 192, 193, 301, 405–408, 411, 416, 428, 430, 432, 457	\Gin@i	195
\endgroup	182, 257, 346	\Gin@ii	197
\endpicture	148	\Gin@llx	227, 231, 332, 337, 405, 426, 495, 526, 548
\endsp@createImage	349	\Gin@lly	227, 235, 246, 332, 337, 406, 424, 495, 526, 548
\endverbatim	164	\Gin@name	227, 228, 254, 314, 332, 338, 359, 361, 405–408, 444, 468, 517, 523, 538, 544
environments:		\Gin@nat@height	423, 424
createImage	300	\Gin@nat@width	425, 426
\eq@driver@name	24	\Gin@nosize	419
\ESO@AeBip@Hook	85, 86, 113, 115, 127, 145, 147	\Gin@postsp	370, 371, 530, 552
\ESO@HookI	103, 113, 124, 145	\Gin@presp	368, 369, 528, 550
\ESO@HookII	103, 104, 113, 114, 125, 145, 146	\Gin@req@height	439, 459
\ESO@HookIII	103, 113, 126, 145	\Gin@req@sizes	427
\ESO@isMEMOIR	88, 131	\Gin@req@width	437, 440, 460
\ESO@yoffsetI	98, 112, 130	\Gin@scalex	478, 481, 485, 488
\expandafter	166, 167, 301, 316, 357, 411, 428, 430, 432, 447, 473	\Gin@scalley	479, 480, 489
		\Gin@setfile	400
		\Gin@SMask	366, 367
		\Gin@tempa	464, 473
		\Gin@transparency	256, 279, 339, 340, 364, 365, 520, 521, 541, 542
		\Gin@transparencygroup	208, 256, 279, 288, 289, 292, 324, 325, 328, 362, 363
F		\Gin@urx	227, 230, 332, 337, 407, 425, 495
\fbox	461	\Gin@ury	227, 234, 245, 332, 337, 408, 423, 495
\fboxsep	461	\Gin@viewport@code	422, 547
\fi	25, 27, 67, 97, 124–127, 139, 151, 243, 249, 250, 267, 287, 293, 294, 322, 329, 330, 340, 390, 392, 394, 401, 415, 417, 420, 421, 433, 434, 444, 448, 456, 461, 462, 475, 486, 490, 491, 499, 510, 515, 522, 536, 543, 549, 555	\Gin@include@eps	360
\filename@area	470, 472	\Gin@include@eps@SP	360, 493
\filename@base	470, 472	\Gin@include@graphics	198
\filename@parse	210	\Gin@input@path	209
\final	12	\Gin@SP@bc	378, 379, 390
		\Gin@SP@group	376, 377, 393, 397, 507
		\Gin@SP@SMask	372, 373, 386, 504, 505, 515, 536
		\Gin@SP@subtype	374, 375, 389
		\Gin@SP@tr	380, 382, 391, 392
		\global	104, 114, 115, 142, 146, 147, 432
		\grcxsp@coverEmbeds	260, 264, 266
		\grcxsp@hideEPS	30, 46
		\grcxsp@importSF	193
		\grcxsp@maxDim	268, 271, 272, 285, 286, 497, 498
		\grcxsp@path0f	192
		\grcxsp@setPictureDimen	169, 226, 331
		\Gread@eps	224
		\Gread@false	401
		\Gread@parse@bb	316
		\group	372
		\grcxsp@createImage	317, 320
		\grcxsp@cnt	168
		\Gscale@div	233, 239, 248
		\gxsp@drivernum	6, 7, 11, 25, 50

\gxsp@embedEPS 254, 268
\gxsp@embedSF 229, 233, 236, 238, 239, 241, 242, 248, 253
\gxsp@embedSFSave 238, 240
\gxsp@Gin@base 224, 254
\gxsp@IWVO 165
\gxsp@literalps@out . 49, 270, 276, 280, 335, 513, 534
\gxsp@psMrk . 47, 271, 276, 296, 297, 336, 340, 342,
386, 389, 521, 523, 524, 529, 542, 544, 545, 551
\gxsp@scaleFactor@x 478, 482, 489, 525, 546
\gxsp@scaleFactor@y 479, 483, 485, 525, 546
\gxsp@setBBox 493, 516, 537
\gxsp@setPSCoor 60, 64, 270, 280, 335, 514, 535
\gxsp@showembeds 8–10, 263

H

\hb@xt@ 437
\hbox 457
\heightOf 184
\hfill 262
\hrule 440, 450
\hss 438, 451
\ht 459

I

\if 263
\if@Ginnamed 358, 404, 442, 453, 467
\if@spx@shownameindraft 19, 443
\if@tempswa 128
\if@twoside 93, 135
\ifdim 232, 237, 247
\IfFileExists 470
\ifGin@bbox 402, 512
\ifGin@clip 281, 494
\ifGin@draft 436
\ifGread@ 403
\ifnum 50
\ifodd 93, 135
\ifpreview 2, 461
\ifspxGin@draft 20, 27
\ifx 24, 124–127, 288, 289, 321, 324,
325, 339, 386, 390, 391, 401, 411, 428, 430,
480, 481, 488, 504, 507, 515, 520, 536, 541, 547
\immediate 160, 165
\includegraphics 275, 357
\input@path 209
\InputIfExists 21
\inputlineno 202, 215, 309
\insertEPS 351

L

\leavevmode 435
\LenToUnit 102, 112
\let 8–10, 46,
65, 66, 104, 114, 115, 143, 146, 147, 157, 191,
208, 209, 266, 319, 348, 349, 354, 355, 360, 432
\llxOf 184, 282
\llyOf 184, 282
\long 320

M

\meaning 447
\message 274, 502
\MessageBreak 202, 203, 215, 218, 219, 302, 303, 309, 310

N

\n 81
\name 358
\newcommand 86, 199, 269, 304, 352, 356, 396
\newcount 168
\newif 2, 19, 20, 358
\noexpand 252–255, 334, 335, 465, 466, 505
\nointerlineskip 100, 110, 149

P

\p@ 240
\PackageError 201, 213, 308, 508
\PackageWarning 300
\paperheight 237, 239, 247, 248, 262
\paperwidth 91, 95, 133, 137, 232, 233, 262
\parbox 262
\PassOptionsToPackage 13, 16
\pictur@ 144
\postEP 319
\postsp 368
\presp 368
\preview 2
\previewfalse 2, 4
\previewOff 4
\previewOn 3
\previewtrue 3, 5
\ProcessOptions 22
\protect 143
\providecommand 3, 4
\ProvidesFile 455
\put 102, 112

R

\relax 46, 50, 143, 194, 266, 351, 411, 428, 430, 547
\renewcommand 121

\RequirePackage	26, 28, 29	\transparency	364
\reserved@a	321	\transparencyGroup	362
\rlap	444, 447	\trimedge	90, 94, 132, 136
		\trimtop	99, 140
S		\ttfamily	444, 447
\scalebox	253		
\setbox	142, 457		
\setkeys	225, 314, 397, 505	U	
\setlength	461	\unitlength	141
\setSMask	385	\urxOf	184, 283
\!showembeds	2	\uryOf	184, 283
\showembeds	2		
\!shownonames	12	V	
\shownonames	12	\vbox	142, 143, 439
\SMask	372	\verbatim@line	161
\SMaskSP@Identity	384, 391	\verbatim@out	160, 165
\SMaskSP@None	381, 383, 386, 507	\verbatim@processline	159
\sp@createImage	304, 348	\verbatim@start	162
\sp@expand@temp	505, 506	\verbatimwrite	156
\sp@setSMask	385, 398, 515, 536	\vfill	262
\spx@sanitize	194, 197	\vrule	438, 451
\spxGin@draftfalse	15, 20	\vss	441, 449
\spxGin@drafttrue	12		
\stockwidth	92, 96, 134, 138	W	
\string	81, 301	\wd	460
\strip@prefix	447	\widthOf	184
\strip@pt	144, 177, 181, 242	\write	160, 165
\subtype	372		
		X	
T		\x@namedef	166, 170–172, 177, 181, 228, 306
\tc	372	\xdef	166
\temp@transparencyGroup	323, 338	\xsp@sanitize	351, 352
\the	161, 202, 215, 309		
\thepage	155	Z	
		\z@	457–461

10 Change History

v.6 (2008/06/15)

createImage: Made `createImage` into a private environment, `sp@createImage`, which can be used anywhere, then created a public version, `createImage`, which is restricted to the preamble. This change is needed to create dynamic appearances in the `rmannot` package. 11

v.7b (2012/09/11)

shownonames: Added `draft`, `final`, `shownonames` options 2

v1.0.2 (2018/11/20)

\csOf: Added `\csOf` 7
createImage: Added convenience command to reference xobjects using commands 11

v1.0.3 (2019/11/13)

General: latex core defines `\IfExists@` which broke this package, we try a fix here. 17

v1.0a (2017/03/12)

showembeds: Added `\previewOn` and `\previewOff` 2