GraphicxSP: Re-using EPS files

D. P. Story
Email: dpstory@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 eso-pic 4
6 Useful Supporting Commands 6
7 The Main Section 6
8 Messing with graphicx 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

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.
(2017/03/12) Added two convenience commands.

\providecommand{\previewOn}{\previewtrue}
\providecommand{\previewOff}{\previewfalse}
\DeclareOption{preview}{\previewtrue}
\DeclareOption{dvipsone}{\def\gxsp@drivernum{0}}
\DeclareOption{dvips}{\def\gxsp@drivernum{1}}
\DeclareOption{showembeds}{\let\gxsp@showembeds=0}
\DeclareOption{!showembeds}{\let\gxsp@showembeds=1}
\let\gxsp@showembeds=1
\def\gxsp@drivernum{1}

The draft mode passes draft on to graphicx. The images appear as rectangles, with the name of the image. The shwononames option removes the name inside the rectangle.

\DeclareOption{draft}{\spxGin@drafttrue
\PassOptionsToPackage{draft}{graphicx}}
\DeclareOption{!draft}{\spxGin@draftfalse
\PassOptionsToPackage{final}{graphicx}}
\DeclareOption{shownonames}{\@spx@shownameindraftfalse}
\DeclareOption{!shownonames}{\@spx@shownameindrafttrue}
\newif\@spx@shownameindraft \@spx@shownameindrafttrue
\newif\ifspxGin@draft \spxGin@draftfalse

\InputIfFileExists{graphics.cfg}{}{}
\ProcessOptions
\@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.
\def\DVIPSONE{dvipsone}\ifx\eq@driver@name\DVIPSONE
\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.
\RequirePackage{graphicx}
\ifspxGin@draft\Gin@drafttrue\fi
\RequirePackage{eso-pic}
\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.
We use either \texttt{dvips} or \texttt{dvipsone} as the driver, in both cases the following is the special we shall use.

\begin{verbatim}
def\gxsp@psMrk{[
}
def\gxsp@literalps@out#1{\special{ps:#1}}
\end{verbatim}

The following are driver dependent definitions. We begin with \texttt{dvips}.

**Dvips driver.**

\begin{verbatim}
\ifnum\gxsp@drivernum=1\relax
When the driver is \texttt{dvips}, we define some postscript procedures to making conversions between \TeX{} and PDF.
\special{userdict begin
    /TeXtoPDF {65536 div DVImag mul} def % sp to pts
    /PDFtoDvips {72.27 div Resolution mul} def % points to dots
    /PDFtoVDvips {72.27 mul Resolution div} def % dots to points
    /HTeXtoDvips {TeXtoPDF PDFtoDvips} def % sp to dots
    /VTeXtoDvips {TeXtoPDF PDFtoVDvips} def end % sp to dots
\end}
\end{verbatim}

The \texttt{cstr} is used to calculate the lower left corner of the bounding box of an EPS file for \texttt{dvips}.

\begin{verbatim}
\special{userdict begin /cstr {currentpoint translate
    1 PDFtoDvips DVImag mul -1 PDFtoDvips DVImag mul scale}def end}
\end{verbatim}

**Dvipsone driver.** The following is code special to the \texttt{dvipsone} driver.
The `undsc1x` is used to calculate the lower left corner of the bounding box of an EPS file for `dvipsone.`

```
64 \def\gxsps@setPSCoor{undsc1x }
65 \let\b@grxsp@Literal\@empty
66 \let\e@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 }
```

## 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{%
```
\AtBeginShipout{% 
\@tempdima=-\trimedge 
\advance\@tempdima-\paperwidth 
\advance\@tempdima\stockwidth 
@if@twoside@ifodd\c@page\else 
\advance\@tempdima2\trimedge 
\advance\@tempdima\paperwidth 
\advance\@tempdima-\stockwidth 
\fi \fi 
\@tempdimb=\ESO@yoffsetI 
\advance\@tempdimb-\trimtop 
\nointerlineskip 
\AtBeginShipoutUpperLeft{% 
\put(\LenToUnit{\@tempdima},\LenToUnit{\@tempdimb}){% 
\ESO@HookIII\ESO@HookI\ESO@HookII 
\global\let\ESO@HookII\@empty 

} % 
}% 
\AtBeginShipout{% 
\nointerlineskip 
\AtBeginShipoutUpperLeft{% 
\put(0,\LenToUnit{\ESO@yoffsetI}){% 
\ESO@HookIII\ESO@HookII\ESO@HookI\ESO@HookII% 
\global\let\ESO@HookII\@empty 
\global\let\ESO@HookI\@empty 

} % 
}% 
}% 
\AtBeginShipout{% 
\nointerlineskip 
\AtBeginShipoutUpperLeft{% 
\put(\LenToUnit{\ESO@yoffsetI},\LenToUnit{\@tempdima}){% 
\ESO@HookIII\ESO@HookII\ESO@HookI\ESO@HookII% 
\global\let\ESO@HookII\@empty 
\global\let\ESO@HookI\@empty 
\global\let\ESO@HookIII\@empty 
\global\let\ESO@Hook\@empty 

} % 
}% 
\If \@ShipoutPicture is defined, we use the old code.
\renewcommand{\@ShipoutPicture}{% 
\bgroup 
\@tempswafalse% 
\ifx\ESO@HookI\@empty\else\@tempswatrue\fi% 
\ifx\ESO@HookII\@empty\else\@tempswatrue\fi% 
\ifx\ESO@HookIII\@empty\else\@tempswatrue\fi% 
\ifx\ESO@Hook\@empty\else\@tempswatrue\fi% 
\if\@tempswa% 
\@tempdima=1in\@tempdimb=-\@tempdima% 
\advance\@tempdimb\ESO@yoffsetI% 
\ESO@isMEMOIR{% 
\advance\@tempdima\trimedge% 
\advance\@tempdima\paperwidth% 
\advance\@tempdima\stockwidth% 
@if@twoside@ifodd\c@page\else\fi% 
\advance\@tempdima-2\trimedge%

6 Useful Supporting Commands

Some standard code that I use in AeB to wrote verbatim tex to a file.
\def\verbatimwrite{\@bsphack
\let\do\@makeother\dospecials
\catcode'\^I=12
\def\verbatim@processline{\immediate\write\verbatim@out
{\the\verbatim@line}}%}
\verbatim@start
\def\endverbatimwrite{\@esphack}
\def\gxsp@IWVO{\immediate\write\verbatim@out}
\def\x@namedef#1{\expandafter\edef\csname #1\endcsname}
\def\e@namedef#1{\expandafter\edef\csname #1\endcsname}

Below is a counter to ensure each name is unique. It is used in \Ginclude@eps@SP.
\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.
\def\grcxsp@setPictureDimen#1#2#3#4#5{%
Calculate the width and height of the EPS. If the lower-left corner is not (0,0), results may not be predictable.

\begingroup
\@tempdima=#4bp
\advance\@tempdima-#2bp
\@tempdima=.99626\@tempdima
\x@namedef{#1widthOf}{\strip@pt\@tempdima}%
\@tempdima=#5bp
\advance\@tempdima-#3bp
\@tempdima=.99626\@tempdima
\x@namedef{#1heightOf}{\strip@pt\@tempdima}%
\endgroup

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

\def\heightOf#1{\csname#1heightOf\endcsname}
\def\widthOf#1{\csname#1widthOf\endcsname}
\def\bboxOf#1{\csname#1BBox\endcsname}
\def\llxOf#1{\csname#1Gin@llx\endcsname}
\def\llyOf#1{\csname#1Gin@lly\endcsname}
\def\urxOf#1{\csname#1Gin@urx\endcsname}
\def\uryOf#1{\csname#1Gin@ury\endcsname}

Use \csOf to expand a name.

\let\csOf\@nameuse

Other internal commands that save info.

\def\grcxsp@pathOf#1{\csname#1path\endcsname}
\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.

\%\def\spx@sanitize{\catcode`\_=12\relax}
\%\def\Gin@i{
\%\ifnextchar[\]
\%\spx@sanitize\Gin@ii
\%\Ginbboxfalse\Ginclude@graphics}

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.

\newcommand{\embedEPS}[3][{}]{
\begin{group}
\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.

\let\input@path\Ginput@path
\filename@parse{#3.eps}
\Gin@getbase{.eps}
\ifundef{Gin@base}
\PackageError{graphicxsp}{}
\PackageError{graphicxsp}{% Graphics file #3 specified on \the\inputlineno was not found%}
\}{% Verify the file exists, is an eps file, is on the latex search path, or is in the current directory.%
\}
\e@namedef{gxsp@Gin@base}{\Gin@base}
\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.

\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.
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, \texttt{\gxsp@embedSF}, needed to embed the file without exceeding the page boundaries.

\begin{verbatim}
\def\gxsp@embedSF{1}\% 
@tempdima=\Gin@urx bp 
\advance\@tempdima-\Gin@llx bp 
@ifdim\@tempdima->\paperwidth
    \Gscale@div\gxsp@embedSF\paperwidth\@tempdima 
    \@tempdima=\Gin@ury bp 
    \advance\@tempdima-\Gin@lly bp 
    \@tempdima=\gxsp@embedSF\@tempdima 
    \ifdim\@tempdima->\paperheight
        \edef\gxsp@embedSFSave{\gxsp@embedSF}\% 
        \Gscale@div\gxsp@embedSF\paperheight\@tempdima 
        \@tempdima=\gxsp@embedSFSave\p@ 
        \@tempdima=\gxsp@embedSF\@tempdima 
        \edef\gxsp@embedSF{\strip@pt\@tempdima}\% 
    \else 
        \@tempdima=\Gin@ury bp 
        \advance\@tempdima\Gin@lly bp 
        \ifdim\@tempdima->\paperheight
            \Gscale@div\gxsp@embedSF\paperheight\@tempdima 
        \fi 
    \fi
\else 
    \@tempdima=\Gin@ury bp 
    \advance\@tempdima\Gin@lly bp 
    \ifdim\@tempdima->\paperheight
        \Gscale@div\gxsp@embedSF\paperheight\@tempdima 
    \fi
\fi
\end{verbatim}

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

\begin{verbatim}
\edef\@tempa{\noexpand\AddToEmbeddedEPSs\noexpand\AtPageLowerLeft\noexpand\scalebox{\gxsp@embedSF}{\noexpand\gxsp@embedEPS{\Gin@base}{\Gin@name}}}\@tempa 
@endgroup
\end{verbatim}

\begin{verbatim}
\@onlypreamble{\embedEPS}
\end{verbatim}

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

\begin{verbatim}
\def\grcxsp@coverEmbeds{% 
    \AddToEmbeddedEPSs\AtPageLowerLeft{\colorbox{white}}{% 
\end{verbatim}
The \texttt{gxsp@embedEPS} command embeds the file, and is called by \texttt{\embedEPS}. It takes three options: (1) the value of \texttt{transparencyGroup}; (2) the EPS path; and (3) the symbolic name for the graphic.

The bounding box \texttt{/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 \texttt{%%BoundingBox} can be deceiving, set the of \texttt{/BBox} to an array with enormous dimensions, the default is \texttt{\grcxsp@maxDim = 5000}. This can be reset to larger value if you are embedding graphics of even more enormous dimensions.

\def\grcxsp@maxDim{5000}

Now, for the \texttt{gxsp@embedEPS} command that embed the graphic between $BP$ and $EP$.

\newcommand{\gxsp@embedEPS}{\[2\]}{\%
\gxsp@literalps@out\{gsave \gxsp@setPSCoor\gxsp@psMrk/BBox [-\grcxsp@maxDim\space-\grcxsp@maxDim\space\grcxsp@maxDim\space\grcxsp@maxDim]/_objdef {Embedded:#2}\%
\message{<Embedding #1>}%
\includegraphics{#1}\%
\gxsp@literalps@out\{\gxsp@psMrk/EP pdfmark\}
\}%

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.

\includegraphics{#1}\%
\gxsp@literalps@out\{\gxsp@psMrk/EP pdfmark\}
\}%

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 \texttt{\embedEPS}

\def\gcxsp@wrapEmbeddedFigure#1#2#3{\%
\def\Gin@transparencygroup{#1}\def\Gin@transparency{#3}\%
\gxsp@literalps@out\{gsave \gxsp@setPSCoor\ifGin@clip\[/BBox \[\llxOf{#2}\space\llyOf{#2}\space\urxOf{#2}\space\uryOf{#2}\space\]/_objdef {#2}\%
\message{<Embedding #1>}%
\includegraphics{#1}\%
\gxsp@literalps@out\{\gxsp@psMrk/EP pdfmark\}
\}%
\%
\def\Gin@wrapEmbeddedFigure#1#2#3{\%
\def\Gin@transparencygroup{#1}\def\Gin@transparency{#3}\%
\gxsp@literalps@out\{gsave \gxsp@setPSCoor\if\Gin@clip\[/BBox \[\llxOf{#2}\space\llyOf{#2}\space\urxOf{#2}\space\uryOf{#2}\space\]/_objdef {#2}\%
\message{<Embedding #1>}%
\includegraphics{#1}\%
\gxsp@literalps@out\{\gxsp@psMrk/EP pdfmark\}
\}%
\%
\def\Gin@transparencygroup{\empty}
\%
\def\Gin@transparency{\empty}
\%
\ifx\Gin@transparencygroup\Gin@exclamation\%
\ifx\Gin@transparency\Gin@exclamation\%
\else
\%
\fi\space/_objdef {#2}\%
\}%
\%
}\%
\}
\%
\}}
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.

\def\ci@undef@msg#1{\PackageWarning{graphicxsp}{The command \expandafter\string\csname #1\endcsname is already defined\MessageBreak choose a different name instead of\MessageBreak'#1'}}
\newcommand{\sp@createImage}[3][{}]{%}
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.

\AddToEmbeddedEPSs{\AtPageLowerLeft{\gxsp@literalps@out{gsave \gxsp@setPSCoor
\gxsp@psMrk/BBox
\[\Gin@llx\space\Gin@lly\space\Gin@urx\space\Gin@ury\]
/\_objdef \{\Gin@name\}\temp@transparencyGroup\space/EP pdfmark
\ifx\Gin@transparency\@empty\else
\gxsp@psMrk\Gin@transparency\space/SetTransparency pdfmark\fi
\#1 ?pdfmark
\gxsp@psMrk/EP pdfmark
grestore}}}
\endgroup

\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}
...

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 \Gincludefile to \GincludefileSP. \Gincludefile is the usual way of handling EPS files, \GincludefileSP is how we are to handle files already embedded. Usage:

\includegraphics[name=AdobeDon,width=1in]{myCoolSelfPic}

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}

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:
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.

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

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.
Use \texttt{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.

We redefine \texttt{Gin@setfile}. If the graphic is named, we salt things with the bounding box parameters.
If the `shownonames` option is taken, we do not show the name of the graphic.

If the current graphic is named, then we don’t need to read the bounding box again or see if it exists again.
If not named, we need to handle it in the usual way.

(2019/11/13) The latest \TeX core defines \texttt{IfFileExists@} which broke this package, we try a fix here.

\begin{verbatim}
\IfFileExists{\filename@area\filename@base#1}{}
{\@ifundefined{IfFileExists@}{}\edef\@filef@und{\filename@area\filename@base#1}}
\Gin@tempa\expandafter\@tempa\@filef@und
\edef\Gin@ext{#1}}{}
\fi
\end{verbatim}

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

\begin{verbatim}
\def\Gin@computeSF{!}
\edef\gxsp@scaleFactor@x{\Gin@scalex}
\edef\gxsp@scaleFactor@y{\Gin@scaley}
\ifx\Gin@scaley!\edef\gxsp@scaleFactor@x{1}\fi
\ifx\Gin@scalex!\edef\gxsp@scaleFactor@y{1}\fi
\end{verbatim}

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

\begin{verbatim}
\def\gxsp@setBBox{\ifGin@clip\[/BBox \[\Gin@llx\space\Gin@lly\space\Gin@urx\space\Gin@ury]\else\[/BBox \[-\grcxsp@maxDim\space\grcxsp@maxDim\space\grcxsp@maxDim\space\grcxsp@maxDim]\fi}
\def\Ginclude@eps@SP#1{\bgroup\ifx\GinSP@SMask\@empty\else\edef\sp@expand@temp{\noexpand\setkeys{GinSP}{\GinSP@SMask}}\sp@expand@temp\ifx\SMaskSP@None\@empty\ifx\GinSP@group\@empty
\message{<#1>}\else\edef\gxsp@setBBox{\[/BBox \[\Gin@llx\space\Gin@lly\space\Gin@urx\space\Gin@ury]\else\[/BBox \[-\grcxsp@maxDim\space\grcxsp@maxDim\space\grcxsp@maxDim\space\grcxsp@maxDim]\fi\endgroup}
\message{<#1>}\else\edef\gxsp@setBBox{\[/BBox \[\Gin@llx\space\Gin@lly\space\Gin@urx\space\Gin@ury]\else\[/BBox \[-\grcxsp@maxDim\space\grcxsp@maxDim\space\grcxsp@maxDim\space\grcxsp@maxDim]\fi\endgroup}
\message{<#1>}\else\edef\gxsp@setBBox{\[/BBox \[\Gin@llx\space\Gin@lly\space\Gin@urx\space\Gin@ury]\else\[/BBox \[-\grcxsp@maxDim\space\grcxsp@maxDim\space\grcxsp@maxDim\space\grcxsp@maxDim]\fi\endgroup}
\message{<#1>}\end{verbatim}

See if the user has specified the \texttt{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.

\begin{verbatim}
\def\Gin@include@eps@SP#1{%
\bgroup\ifx\Gin@SMask\@empty\else\edef\sp@expand@temp{\noexpand\setkeys{Gin@}{\Gin@SMask}}\sp@expand@temp\ifx\SMask@None\@empty\else\edef\gxsp@setBBox{\[/BBox \[\Gin@llx\space\Gin@lly\space\Gin@urx\space\Gin@ury]\else\[/BBox \[-\grcxsp@maxDim\space\grcxsp@maxDim\space\grcxsp@maxDim\space\grcxsp@maxDim]\fi\endgroup}
\message{<#1>}\else\edef\gxsp@setBBox{\[/BBox \[\Gin@llx\space\Gin@lly\space\Gin@urx\space\Gin@ury]\else\[/BBox \[-\grcxsp@maxDim\space\grcxsp@maxDim\space\grcxsp@maxDim\space\grcxsp@maxDim]\fi\endgroup}
\message{<#1>}\end{verbatim}

17
If SMask is specified, we call \psSetSMask to set the graphics state parameters for a soft mask.

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

If SMask is specified, we call \psSetSMask to set the graphics state parameters for a soft mask.
\makeatletter\makeatother
\begin{Verbatim}
\begin{verbatim}
\Gin@llx \space \Gin@lly \space moveto \\
\currentpoint translate \fi \\
\Gin@resp \\
\gxsp@psMrk{\//graphicx pregnant} /SP pdfmark \\
\Gin@postsp \\
grestore \\
\}% \\
\fi \\
\egroup \}
\end{verbatim}
\end{Verbatim}
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.

<table>
<thead>
<tr>
<th>Symbols</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>@Ginnamedfalse</td>
<td>358</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>@Ginnamedtrue</td>
<td>360</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>@ShipoutPicture</td>
<td>121</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>@addtofilelist</td>
<td>454</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>@badend</td>
<td>322</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>@bashack</td>
<td>156</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>@cc1v</td>
<td>142, 150</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>@currentvir</td>
<td>321</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>@ehc</td>
<td>431</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>@esphack</td>
<td>164</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>@filef@und</td>
<td>472, 473</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>@gtempa</td>
<td>315, 316</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>@ifnextchar</td>
<td>196</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>@ifpackageloaded</td>
<td>46</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>@ifstar</td>
<td>353</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>@ifundefined</td>
<td>2, 23, 87, 200, 212, 305, 307, 471</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>@latex@error</td>
<td>431</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>@makeother</td>
<td>157</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>@nameuse</td>
<td>191</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>@onlypreamble</td>
<td>299, 350</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>@spx@shownameindraftfalse</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>@spx@shownameindrafttrue</td>
<td>18, 19</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>@tempa</td>
<td>251, 256, 333, 345, 446, 447, 465, 473, 477, 480, 481, 488</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>@tempdima11</td>
<td>98, 99, 102, 129, 130, 140, 144</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>@tempsvafalse</td>
<td>123</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>@tempsvartrue</td>
<td>124–127</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>@width</td>
<td>440</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>_</td>
<td>316, 401</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>_</td>
<td>158</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>_</td>
<td>194, 351</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\active</td>
<td>158</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\AddToEmbeddedEPS</td>
<td>86, 252, 261, 334</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\AddToShipoutPicture</td>
<td>155</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\AtBeginDocument</td>
<td>264</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\AtBeginDvi</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\AtBeginShipout</td>
<td>89, 109</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\AtBeginShipoutUpperLeft</td>
<td>101, 111</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\AtPageCenter</td>
<td>255</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\AtPageLowerLeft</td>
<td>252, 261, 334</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\b@grxsp@Literal</td>
<td>61, 65, 68</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\bbox@f</td>
<td>186</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\bc</td>
<td>372</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\begin{group}</td>
<td>173, 207, 317</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\begin{group}</td>
<td>122, 352, 503</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\box</td>
<td>150, 461</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\c@page</td>
<td>93, 135</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\catcode</td>
<td>158, 194, 351</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\ci@undefinedmsg</td>
<td>300, 305</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\colorbox</td>
<td>261</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\createImage (environment)</td>
<td>348, 350</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\csname</td>
<td>166, 167, 184–190, 192, 193, 301, 405–408, 410, 411, 428, 430, 432, 457</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\csgrf</td>
<td>184</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\DeclareOption</td>
<td>5–9, 12, 14, 15, 17, 18</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\define@key</td>
<td>359, 362, 364, 366, 368, 370, 372, 374, 376, 378, 380, 381</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\do</td>
<td>157</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\dospecials</td>
<td>157</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\dp</td>
<td>458</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\draft</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\dvips</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\dviPSone</td>
<td>24</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\dvipsone</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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 \IfFileExists@ which broke this package, we try a fix here. 17

v1.0a (2017/03/12)
showembeds: Added \previewOn and \previewOff .............................. 2