The **bmpsize** package

Heiko Oberdiek*

<heiko.oberdiek at googlemail.com>

2016/05/16 v1.7

Abstract

Package bmpsize analyzes bitmap images to extract size and resolution data. It adds this feature to the graphics package that now do not need separate bounding box files for bitmap images. Additionally the implementation for the inclusion of bitmap images in some drivers of package graphicx are rewritten to support options viewport, trim and clip.

Contents

---

*Please report any issues at https://github.com/ho-tex/oberdiek/issues

1 Documentation

1.1 Introduction

The support of bitmap images in the TEX world is quite poor. TEX can read text files and thus parse the bounding box of EPS files, but it cannot read binary files. If TEX reads a line, it removes spaces before the line end and normalizes the line end itself to get independent from the convention of the operating system.

The situation changed with pdfTEX. It is a TEX compiler, where the output driver is already integrated. Images of type JPEG and PNG are supported directly and the size of the images are reported back to the TEX language. Thus it is easy for package graphics to get the size of the images.

The problem remains for other drivers than pdfTEX in PDF mode. The size information must either be given manually by the bounding box options or an additional file is used for each image, where the size information is stored as EPS bounding box. Program dvips comes with the program ebb that create these .bb files. However it ignores the natural size of the image and uses a fixed resolution of 100 DPI.

Since pdfTEX 1.30.0 there are some new primites. Especially \pdffiledump is very helpful. It reads a file in binary mode and reports the selected area as hex dump. It works in both DVI and PDF mode of pdfTEX. Thus it is now possible to read and parse bitmap files to get their size. This project uses this feature to implement parsers for many bitmap file types.

1.2 Bitmap image parsers

This project supports the following image types:

> BMP, GIF, JPEG, MSP, PAM, PCX, PNG, PNM, SGI, TGA, TIFF, WMF, XPM

Consult the documentation of your TeX distribution and driver which types are supported by your driver. Sometimes automatically triggered conversions can be configured to extend the range of supported image types.

1.2.1 User interface

Package bmpsize hooks into package graphics. If an image is included and its size is not given, then bmpsize investigates the image. If it could be parsed as known bitmap file type, the size is reported back to package graphics.

The following options are added to the options of package graphicx:

resolutionunit: Specifies the unit of the options for setting the resolution. Default is 1in that means the numbers are interpreted as dots per inch (DPI).

defaultresolution: Bitmap files do not always provide information about their resolution (density). If this information is not given, the values of this option are used to calculate the image size. Default: 72 !

resolution: This option override the resolution given in the bitmap file.

bmpsizefast: Values are true and false. The option is enabled by default. Then mainly ε-TeX's arithmetic is used to calculate the width and height. However the dimen dimensions are limited. Therefore overflow errors can happen. Disable then this option to use the arithmetic of package fp. It allows a larger range of numbers at the cost of speed.

Options defaultresolution and resolution expect two numbers, separated by a space. The first is taken as density for the horizontal x axis, the second for the vertical y axis. One of the numbers may be replaced by an exclamation mark. In this an aspect ratio is respected and the correct density for this axis automatically calculated. If one number is given, this number is used for both axes. Examples:

```
defaultresolution=72 !      % Default
resolution=100              % Simulates behaviour of program ebb
```

The options can be set in \includegraphics or using \bmpsizesetup. \setkeys{Gin} is equivalent to the latter case.

```
\bmpsizesetup{resolutionunit=1in, resolution=100}
\includegraphics[
  defaultresolution=72 !,
  bmpsizefast=false
]{image}
```

1.2.2 Hints

- My version of dvips.def 1999/02/16 v3.0i defines rules for the supported bitmap extensions, but does not include them in the list of extensions that are tried if the file name is not given with an extension. In such a case, the list of extensions can be set by \DeclareGraphicsExtensions, see grfguide. The following code just extends the list:

```
\makeatletter
\g@addto@macro\Gin@extensions{,.bmp,.pcx,.msp}
\makeatother
```

- My version of `dvipdfm.def` 1998/11/24 vx.x misses the graphics rule for PNG files. It can be added by:

  `\DeclareGraphicsRule{.png}{bmp}{.bb}{#1}`

  See the previous issue to add the extension `.png` to the list of extensions for package `graphics`.

1.2.3 Test program

There is a test program `bmpsize-test.tex`. Run it through `latex`, `pdflatex`, or `pdftex`. Then given image files are inspected and the result is printed.

1.2.4 Interface for programmers

The macro names of the parsers are `\bmpsize@read@`⟨*type*⟩. Example: `\bmpsize@read@jpg` in case of JPEG.

A parser sets the switch `\ifbmpsize@ok` to true, if it could successfully parse the image file. The width and height are returnd in `\bmpsize@width` and `\bmpsize@height`. If information about density is available, it is used to calculate width and height of the image, otherwise the values given by option `defaultresolution` is used. `resolution` overwrites the values in the image file.

1.3 Improved bitmap inclusion

Some drivers for package `graphics` define the graphics type `bmp` for bitmap images. The code in the standard drivers for `dvips`, `dvipdfm`, and `dvipdfmx` is very basic and misses essential features of the package `graphicx`. Therefore the code for bitmap inclusion is automatically rewritten by this package to add the following features:

- Support for `viewport` and `trim`.

- Support for `clip`.

- In case of `dvipdfm` and `dvipdfmx` the bitmap images are reused and not included again if they are used more than once.

However, there is a difference between `dvipdfm` and `dvipdfmx`, especially if images are reused. In the former case the reused box has width and height of 1bp, in the latter case its natural width. Thus the correct driver option must be given. `dvipdfm` and `dvipdfmx` are not equivalent.

Older versions of `dvipdfmx` uses a size of 1in. However I do want to distinguish between versions of the same program. Therefore the support of these older versions has stopped with version 1.6 of this package. Use version dvipdfmx-20090708 or newer (some few versions before will probably also work, but I don't want to investigate this further).

2 Implementation

2.1 Basic package **bmpsize-base**

Identification.

```
1 ⟨*base⟩
2 \ProvidesPackage{bmpsize-base}%
3   [2016/05/16 v1.7 Basic part of bmpsize (HO)]%
```

Modules of package `fp` are used for calculations.

```
4 \RequirePackage{fp-basic}
5 \RequirePackage{fp-snap}
```

Package `fp` uses nested `\loop` structures. That breaks with the plain-TeX version of `\loop`. Therefore we use the LaTeX variant.

```
 6 \long\def\@bmpsize@plain@loop#1\repeat{%
 7   \def\iterate{%
 8     #1\relax
 9     \expandafter\iterate\fi
10   }%
11   \iterate
12   \let\iterate\relax
13 }
14 \RequirePackage{pdftexcmds}[2007/11/11]
15 \newif\ifbmpsize@ok
16 \let\@bmpsize@ok\bmpsize@oktrue
17
18 \newif\if@bmpsize@bigendian
19 \newif\if@bmpsize@absnum
20 \newif\if@bmpsize@user@resolution
21 \newif\if@bmpsize@fast
22 \@bmpsize@fasttrue
23
24 \def\@bmpsize@init{%
25   \let\@bmpsize@org@plain@loop\loop
26   \let\loop\@bmpsize@plain@loop
27   \bmpsize@okfalse
28   \@bmpsize@bigendiantrue
29   \@bmpsize@absnumfalse
30   \let\bmpsize@pixelwidth\relax
31   \let\bmpsize@pixelheight\relax
32   \let\bmpsize@pixelx\relax
33   \let\bmpsize@pixely\relax
34   \let\bmpsize@unit\relax
35   \let\bmpsize@pixelxdenom\relax
36   \let\bmpsize@pixelydenom\relax
37   \let\bmpsize@orientation\relax
38 }
39
40 \def\@bmpsize@stop#1\@nil{}
41
42 \def\@bmpsize@loop#1{%
43   #1%
44   \@bmpsize@loop{#1}%
45 }
46 \def\@bmpsize@break#1\@bmpsize@loop#2{}
47
48 \def\@bmpsize@size#1#2#3{%
49   \edef#3{\pdf@filesize{#1}}%
50   \ifx#3\@empty
51     \expandafter\@bmpsize@stop
52   \fi
53   \ifnum#3<#2\relax
54     \expandafter\@bmpsize@stop
55   \fi
56 }
57
58 \def\@bmpsize@read#1#2#3{%
59   \edef\@bmpsize@buf{\pdf@filedump{#3}{#2}{#1}}%
60   \edef\@bmpsize@temp{%
61     \noexpand\@bmpsize@check@byte{#2}\@bmpsize@buf{}{}\noexpand\\%
62   }%
63   \@bmpsize@temp
64 }
65 \def\@bmpsize@fillbuf#1{%
66   \ifx\@bmpsize@buf\@empty
```

```
67    \expandafter\@firstofone
68  \else
69    \expandafter\@gobble
70  \fi
71  {%
72    \edef\@bmpsize@buf{%
73      \pdf@filedump{\bmpsize@offset}{\bmpsize@fillbuflength}{#1}%
74    }%
75    \ifx\@bmpsize@buf\@empty
76      \expandafter\@bmpsize@stop
77    \fi
78    \edef\bmpsize@offset{\the\numexpr\bmpsize@offset+\bmpsize@fillbuflength}%
79  }%
80 }
81 \def\bmpsize@fillbuflength{10}
82
83 \def\@bmpsize@append#1#2#3{%
84   \edef#1{#2#3}%
85 }
86 \def\@bmpsize@pushback#1{%
87   \edef\@bmpsize@buf{#1\@bmpsize@buf}%
88 }
89
90 \def\@bmpsize@iswhite#1{%
91   \ifnum\pdf@strcmp{#1}{09}=\z@
92   \else
93     \ifnum\pdf@strcmp{#1}{0A}=\z@
94     \else
95       \ifnum\pdf@strcmp{#1}{0D}=\z@
96       \else
97         \ifnum\pdf@strcmp{#1}{20}=\z@
98         \else
99           1%
100        \fi
101      \fi
102    \fi
103  \fi
104  \space
105 }
106 \def\@bmpsize@isdigit#1{%
107   \ifnum\pdf@strcmp{#1}{30}<\z@
108     1%
109   \else
110     \ifnum\pdf@strcmp{#1}{39}>\z@
111       1%
112     \fi
113   \fi
114   \space
115 }
116
117 \def\@bmpsize@check@byte#1#2#3{%
118   \ifnum#1<\@ne
119     \csname fi\endcsname
120     \@bmpsize@cleanup@end
121   \else
122     \csname fi\endcsname
123   \ifx!#2#3!%
124     \csname fi\endcsname
125     \@bmpsize@stop
126   \else
127     \csname fi\endcsname
128     \expandafter\@bmpsize@check@byte\expandafter{\the\numexpr#1-1}%
```

```
129 }
130 \def\@bmpsize@cleanup@end#1\\{}
131
132 \def\@bmpsize@swap@maybe#1{%
133  \if@bmpsize@bigendian
134  \else
135   \edef#1{\expandafter\@bmpsize@@swap#1\@empty\@empty\@empty\@empty}%
136  \fi
137 }
138 \def\@bmpsize@@swap#1#2#3#4#5#6#7#8{%
139  #7#8#5#6#3#4#1#2%
140 }
141
142 \def\@bmpsize@skip@one{%
143  \edef\@bmpsize@buf{\expandafter\@gobbletwo\@bmpsize@buf}%
144 }
145 \def\@bmpsize@skip@two{%
146  \edef\@bmpsize@buf{\expandafter\@gobblefour\@bmpsize@buf}%
147 }
148 \def\@bmpsize@skip@four{%
149  \edef\@bmpsize@buf{%
150   \expandafter\expandafter\expandafter\@gobblefour\expandafter
151   \@gobblefour\@bmpsize@buf
152  }%
153 }
154
155 \def\@bmpsize@grab#1#2{%
156  \edef#1{\noexpand\@bmpsize@grab@byte#2=\@bmpsize@buf\noexpand\\}%
157  \edef#1{#1}%
158 }
159 \def\@bmpsize@grab@byte#1=#2#3{%
160  #2#3%
161  \ifnum#1>\@ne
162   \expandafter\@bmpsize@grab@byte\the\numexpr#1-1\expandafter=%
163  \else
164   \expandafter\@bmpsize@cleanup@end
165  \fi
166 }
167
168 \def\@bmpsize@abs@maybe#1{%
169  \let\@bmpsize@temp\relax
170  \if@bmpsize@absnum
171   \ifnum"\expandafter\@car#1\@nil>7 %
172    \edef#1{\expandafter\@bmpsize@abs@byte#1\relax}%
173    \ifnum\pdf@strcmp{#1}{7FFFFFFF}=\z@
174     \let\@bmpsize@temp\@bmpsize@stop
175    \else
176     \def\@bmpsize@temp{\edef#1{\the\numexpr#1+1}}%
177    \fi
178   \fi
179  \fi
180 }
181 \def\@bmpsize@abs@byte#1{%
182  \ifx#1\relax
183  \else
184   \ifcase"0#1 %
185    F\or E\or D\or C\or B\or A\or 9\or 8\or
186    7\or 6\or 5\or 4\or 3\or 2\or 1\or 0%
187   \fi
188   \expandafter\@bmpsize@abs@byte
189  \fi
190 }
```

```
191
192 \def\@bmpsize@num@one#1{%
193  \@bmpsize@grab#11%
194  \@bmpsize@abs@maybe#1%
195  \edef#1{\number"#1}%
196  \@bmpsize@temp
197  \@bmpsize@skip@one
198 }
199 \def\@bmpsize@num@two#1{%
200  \@bmpsize@grab#12%
201  \@bmpsize@swap@maybe#1%
202  \@bmpsize@abs@maybe#1%
203  \edef#1{\number"#1}%
204  \@bmpsize@temp
205  \@bmpsize@skip@two
206 }
207 \def\@bmpsize@num@four#1{%
208  \@bmpsize@grab#14%
209  \@bmpsize@swap@maybe#1%
210  \@bmpsize@abs@maybe#1%
211  \ifnum\pdf@strcmp{#1}{7FFFFFFF}>\z@
212    \expandafter\@bmpsize@stop
213  \fi
214  \edef#1{\number"#1}%
215  \@bmpsize@temp
216  \@bmpsize@skip@four
217 }
218
219 \def\@bmpsize@div#1#2#3{% #1 := #2/#3
220  \FPdiv#1{#2}{#3}%
221  \@bmpsize@beautify#1%
222 }
223 \def\@bmpsize@beautify#1{%
224  \FPifint#1%
225    \edef#1{\expandafter\@bmpsize@trunc#1.\@nil}%
226  \else
227    \edef#1{\expandafter\@bmpsize@cleanup@frac#1.\@nil}%
228  \fi
229 }
230 \def\@bmpsize@trunc#1.#2\@nil{#1}
231 % #1 isn't an integer, thus we should have at least one
232 % necessary digit after the dot
233 \def\@bmpsize@cleanup@frac#1.#2#3.#4\@nil{%
234  #1.#2%
235  \ifx\\#3\\%
236  \else
237    \@bmpsize@cleanup@fracdigits#3000000000\@nil
238  \fi
239 }
240 \def\@bmpsize@cleanup@fracdigits#1#2#3#4#5#6#7#8#9{%
241  \ifcase#9 %
242    \ifcase#8 %
243      \ifcase#7 %
244        \ifcase#6 %
245          \ifcase#5 %
246            \ifcase #4 %
247              \ifcase #3 %
248                \ifcase #2 %
249                  \ifcase #1 %
250                  \else
251                    #1%
252                  \fi
```

8

```
253        \else
254          #1#2%
255        \fi
256        \else
257         #1#2#3%
258        \fi
259       \else
260        #1#2#3#4%
261       \fi
262      \else
263       #1#2#3#4#5%
264      \fi
265     \else
266      #1#2#3#4#5#6%
267     \fi
268    \else
269     #1#2#3#4#5#6#7%
270    \fi
271   \else
272    #1#2#3#4#5#6#7#8%
273   \fi
274  \else
275   #1#2#3#4#5#6#7#8#9%
276  \fi
277  \@bmpsize@trunc.%
278 }
279
280 \def\@bmpsize@end{%
281  \ifbmpsize@ok
282   \ifx\bmpsize@pixelwidth\relax
283    \bmpsize@okfalse
284   \fi
285   \ifx\bmpsize@pixelheight\relax
286    \bmpsize@okfalse
287   \fi
288  \fi
289  \ifbmpsize@ok
290   \ifnum\bmpsize@pixelwidth>\z@
291   \else
292    \bmpsize@okfalse
293   \fi
294   \ifnum\bmpsize@pixelheight>\z@
295   \else
296    \bmpsize@okfalse
297   \fi
298  \fi
299  \ifbmpsize@ok
300   \ifcase 0%
301    \ifx\bmpsize@pixelx\relax 1 \fi
302    \ifx\bmpsize@pixely\relax 1 \fi
303    \ifnum\bmpsize@pixelx>\z@\else 1 \fi
304    \ifnum\bmpsize@pixely>\z@\else 1 \fi
305    \ifx\bmpsize@pixelxdenom\relax
306     \ifx\bmpsize@pixelydenom\relax\else 1 \fi
307    \else
308     \ifnum\bmpsize@pixelxdenom>\z@\else 1 \fi
309    \fi
310    \ifx\bmpsize@pixelydenom\relax
311    \else
312     \ifnum\bmpsize@pixelydenom>\z@\else 1 \fi
313    \fi
314   \else
```

```
315    \let\bmpsize@pixelx\relax
316    \let\bmpsize@pixely\relax
317    \let\bmpsize@unit\relax
318    \let\bmpsize@pixelxdenom\relax
319    \let\bmpsize@pixelydenom\relax
320   \fi
321   \ifx\bmpsize@pixelxdenom\relax
322   \else
323    \@bmpsize@div\bmpsize@pixelx\bmpsize@pixelx\bmpsize@pixelxdenom
324    \@bmpsize@div\bmpsize@pixely\bmpsize@pixely\bmpsize@pixelydenom
325    \let\bmpsize@pixelxdenom\relax
326    \let\bmpsize@pixelydenom\relax
327   \fi
328   \ifcase 0\ifx\bmpsize@unit\relax 1\fi
329         \if@bmpsize@user@resolution 1\fi
330         \relax
331    \let\bmpsize@calc@unit\bmpsize@unit
332    \let\bmpsize@calc@pixelx\bmpsize@pixelx
333    \let\bmpsize@calc@pixely\bmpsize@pixely
334   \else
335    \let\bmpsize@calc@unit\bmpsize@unit@default
336    \let\bmpsize@calc@pixelx\bmpsize@pixelx@default
337    \let\bmpsize@calc@pixely\bmpsize@pixely@default
338    \ifx\bmpsize@calc@pixely\Gin@exclamation
339     \ifx\bmpsize@pixelx\relax
340      \let\bmpsize@calc@pixely\bmpsize@calc@pixelx
341     \else
342      \FPdiv\bmpsize@calc@pixely\bmpsize@calc@pixelx\bmpsize@pixelx
343      \FPmul\bmpsize@calc@pixely\bmpsize@calc@pixely\bmpsize@pixely
344     \fi
345    \else
346     \ifx\bmpsize@calc@pixelx\Gin@exclamation
347      \ifx\bmpsize@pixelx\relax
348       \let\bmpsize@calc@pixelx\bmpsize@calc@pixely
349      \else
350       \FPdiv\bmpsize@calc@pixelx\bmpsize@calc@pixely\bmpsize@pixely
351       \FPmul\bmpsize@calc@pixelx\bmpsize@calc@pixelx\bmpsize@pixelx
352      \fi
353     \fi
354    \fi
355   \fi
356   \FPdiv\bmpsize@width\bmpsize@pixelwidth\bmpsize@calc@pixelx
357   \FPdiv\bmpsize@height\bmpsize@pixelheight\bmpsize@calc@pixely
358   % calculation of width and height in bp for package graphics
359   % 1in = 72bp = 72.27pt, 72/72.27 = 8/8.03, 1pt = 65536sp
360   \if@bmpsize@fast
361    \edef\bmpsize@width{%
362     \strip@pt\dimexpr.99626\dimexpr
363     \bmpsize@width\dimexpr\bmpsize@calc@unit
364    }%
365    \edef\bmpsize@height{%
366     \strip@pt\dimexpr.99626\dimexpr
367     \bmpsize@height\dimexpr\bmpsize@calc@unit
368    }%
369   \else
370    \edef\@bmpsize@temp{\number\dimexpr\bmpsize@calc@unit}%
371    \ifnum\@bmpsize@temp>100000 %
372     \FPmul\@bmpsize@temp\@bmpsize@temp{0.00001}%
373     \def\@bmpsize@corr{100000}%
374    \else
375     \let\@bmpsize@corr\relax
376    \fi
```

```
377     \FPmul\bmpsize@width\bmpsize@width\@bmpsize@temp
378     \FPmul\bmpsize@height\bmpsize@height\@bmpsize@temp
379     \FPmul\bmpsize@width\bmpsize@width{8}%
380     \FPmul\bmpsize@height\bmpsize@height{8}%
381     \FPdiv\bmpsize@width\bmpsize@width{8.03}%
382     \FPdiv\bmpsize@height\bmpsize@height{8.03}%
383     \FPdiv\bmpsize@width\bmpsize@width{65536}%
384     \FPdiv\bmpsize@height\bmpsize@height{65536}%
385     \ifx\@bmpsize@corr\relax
386     \else
387      \FPmul\bmpsize@width\bmpsize@width\@bmpsize@corr
388      \FPmul\bmpsize@height\bmpsize@height\@bmpsize@corr
389     \fi
390     \FPround\bmpsize@width\bmpsize@width{5}%
391     \FPround\bmpsize@height\bmpsize@height{5}%
392     \@bmpsize@beautify\bmpsize@width
393     \@bmpsize@beautify\bmpsize@height
394    \fi
395   \fi
396   \let\loop\@bmpsize@org@plain@loop
397 }
398 \def\bmpsize@unit@default{72.27pt}% more accurate than 1in
399 \def\bmpsize@pixelx@default{72}
400 \let\bmpsize@pixely@default\Gin@exclamation
401
402 \def\bmpsize@types{png,jpg,bmp,gif,tiff,pnm,pam,xpm,tga,pcx,msp,sgi}
403 ⟨/base⟩
```

2.2 Bitmap formats

2.2.1 png

```
begin png
big-endian

read 24 0
grab 8       -> $temp
check streq $temp [0x89 "PNG" 0x0D 0x0A 0x1A 0x0A]
num 4        -> $length
grab 4       -> $temp
check streq $temp ["IHDR"]
num 4        -> $pixelwidth
num 4        -> $pixelheight
ok
assign numexpr(20 + $length) -> $offset
loop
 read 8 $offset
 num 4       -> $length
 grab 4      -> $temp
 if streq $temp ["IDAT"]
   stop
 fi
 if streq $temp ["pHYs"]
   read 9 numexpr($offset + 8)
   num 4     -> $pixelx
   num 4     -> $pixely
   grab 1    -> $temp
   if numeq $temp 1
     assign {100cm} -> $unit
   fi
   stop
 fi
 assign numexpr($offset + 12 + $length) -> $offset
```

```
                    repeat
                    end
```

\bmpsize@read@png

```
404 ⟨*base⟩
405 \def\bmpsize@read@png#1{%
406   \@bmpsize@init
407   \@bmpsize@bigendiantrue
408   \@bmpsize@read{#1}{24}{0}%
409   \@bmpsize@grab\bmpsize@temp{8}%
410   \@bmpsize@skip@four
411   \@bmpsize@skip@four
412   \ifnum\pdf@strcmp{\bmpsize@temp}{89504E470D0A1A0A}=\z@
413   \else
414     \expandafter\@bmpsize@stop
415   \fi
416   \@bmpsize@num@four\bmpsize@length
417   \@bmpsize@grab\bmpsize@temp{4}%
418   \@bmpsize@skip@four
419   \ifnum\pdf@strcmp{\bmpsize@temp}{49484452}=\z@
420   \else
421     \expandafter\@bmpsize@stop
422   \fi
423   \@bmpsize@num@four\bmpsize@pixelwidth
424   \@bmpsize@num@four\bmpsize@pixelheight
425   \@bmpsize@ok
426   \edef\bmpsize@offset{\the\numexpr20+\bmpsize@length}%
427   \@bmpsize@loop{%
428     \@bmpsize@read{#1}{8}{\bmpsize@offset}%
429     \@bmpsize@num@four\bmpsize@length
430     \@bmpsize@grab\bmpsize@temp{4}%
431     \@bmpsize@skip@four
432     \ifnum\pdf@strcmp{\bmpsize@temp}{49444154}=\z@
433       \expandafter\@firstofone
434     \else
435       \expandafter\@gobble
436     \fi
437     {%
438       \@bmpsize@stop
439     }%
440     \ifnum\pdf@strcmp{\bmpsize@temp}{70485973}=\z@
441       \expandafter\@firstofone
442     \else
443       \expandafter\@gobble
444     \fi
445     {%
446       \@bmpsize@read{#1}{9}{\numexpr\bmpsize@offset+8\relax}%
447       \@bmpsize@num@four\bmpsize@pixelx
448       \@bmpsize@num@four\bmpsize@pixely
449       \@bmpsize@grab\bmpsize@temp{1}%
450       \@bmpsize@skip@one
451       \ifnum\bmpsize@temp=1\relax
452         \expandafter\@firstofone
453       \else
454         \expandafter\@gobble
455       \fi
456       {%
457         \def\bmpsize@unit{100cm}%
458       }%
459       \@bmpsize@stop
460     }%
461     \edef\bmpsize@offset{\the\numexpr\bmpsize@offset+12+\bmpsize@length}%
462 }%
```

```
463  \@bmpsize@stop
464  \@nil
465  \@bmpsize@end
466 }%
467 ⟨/base⟩
```

2.2.2 jpg

begin jpg

```
read 3 0
grab 3      -> $temp % SOI and 0xFF
check streq $temp [0xFF 0xD8 0xFF]
assign {2} -> $offset
assign {0} -> $exifdensity
loop
  read 4 $offset
  grab 1    -> $temp
  check streq $temp [0xFF]
  num 1     -> $temp
  if numeq $temp 0xDA % SOS
    stop
  fi
  % look for JFIF APP0 segment
  if numeq $temp 0xE0 % APP0
    num 2      -> $length
    if numeq $exifdensity 0
      if numge $length 16 % a JFIF segment has 16 bytes at least
        read 12 numexpr($offset + 4)
        grab 5     -> $temp % identifier
        if streq $temp ["JFIF" 0x0]
          check numge $length 16
          skip 2 % version
          num 1       -> $temp % units
          if numeq $temp 1
            assign {72.27pt} -> $unit
          else
            if numeq $temp 2
              assign {1cm} -> $unit
            fi
          fi
          num 2   -> $pixelx
          num 2   -> $pixely
        fi
      fi
    fi
  else
    if numeq $temp 0xE1 % APP1
      % look for Exif APP1 segment
      num 2 -> $length
      if numge $length 20 % identifier (6) + Tiff header (8) + first IFD (>=6)
        read 20 numexpr($offset + 4)
        grab 6 -> $temp
        if streq $temp ["Exif" 0x0 0x0]
          assign numexpr($offset + 10) -> $exifoffset
          % read TIFF header
          grab 2 -> $temp
          if streq $temp ["II"]
            little-endian
          else
            check streq $temp ["MM"]
            % big-endian
          fi
```

13

```
num 2 -> $temp
check numeq $temp 42
num 4 -> $temp % offset of first IFD
check numgt $temp 0
% read first IFD
assign numexpr($temp + $exifoffset) -> $off
read 2 $off
num 2 -> $entries
assign numexpr($off + 2) -> $off
loop
  if numeq $entries 0
    break
  fi
  assign numexpr($entries - 1) -> $entries
  % entry format:
  % 2 tag
  % 2 field type
  % 4 count
  % 4 value/offset
  read 12 $off
  assign numexpr($off + 12) -> $off
  num 2 -> $tag
  if numeq $tag 296 % ResolutionUnit
    skip 6 % type: 3 (short), count: 1
    num 2 -> $temp
    ifcase $temp
    or % 1
      clear $unit
    or % 2
      assign {72.27pt} -> $unit
    or % 3
      assign {1cm} -> $unit
    else
      clear $unit % unknown
    fi
    ifcase $temp
    or % 1
    or % 2
      assign {1} -> $exifdensity
    or % 3
      assign {1} -> $exifdensity
    else
      assign $exifdensity -> $exifdensity
    fi
  fi
  % 256 ImageWidth (use width of JPG part)
  % 257 ImageHeight (use height of JPG part)
  if numeq $tag 274 % Orientation
    skip 6 % type: 3 (short), count: 1
    num 2 -> $temp
    if numge $temp 0
      if numle $temp 8
        assign $temp -> $orientation
      fi
    fi
  fi
  if numeq $tag 282 % XResolution
    skip 6
    num 4 -> $temp
    read 8 numexpr($temp + $exifoffset)
    num 4 -> $pixelx
    num 4 -> $temp
```

14

```
            if numeq $temp 1
            else
              assign numexpr($temp) -> $pixelxdenom
              % div $pixelx $temp -> $pixelx
            fi
          fi
        if numeq $tag 283 % YResolution
          skip 6
          num 4 -> $temp
          read 8 numexpr($temp + $exifoffset)
          num 4 -> $pixely
          num 4 -> $temp
          if numeq $temp 1
          else
            assign numexpr($temp) -> $pixelydenom
            % div $pixely $temp -> $pixely
          fi
        fi
      repeat
      big-endian
    fi
  fi
else
  assign numexpr($temp - 0xC0) -> $temp
  ifcase $temp % SOF_0
  or % SOF_1
  or % SOF_2
  or % SOF_3
  or % DHT
    assign {-1} -> $temp
  or % SOF_5
  or % SOF_6
  or % SOF_7
  or % JPG
    assign {-1} -> $temp
  or % SOF_9
  or % SOF_10
  or % SOF_11
  or % DAC
    assign {-1} -> $temp
  or % SOF_13
  or % SOF_14
  or % SOF_15
  else
    assign {-1} -> $temp
  fi
  if numeq $temp -1
  else
    read 4 numexpr($offset + 5)
    num 2  -> $pixelheight
    num 2  -> $pixelwidth
    if numeq $pixelheight 0
      clear $pixelheight
      stop
    fi
    ok
    stop
  fi
  num 2 -> $length
fi
fi
assign numexpr($offset + $length + 2) -> $offset
```

```
                    repeat
                    end
```

`\bmpsize@read@jpg`

```
468 ⟨*base⟩
469 \def\bmpsize@read@jpg#1{%
470   \@bmpsize@init
471   \@bmpsize@read{#1}{3}{0}%
472   \@bmpsize@grab\bmpsize@temp{3}%
473   \@bmpsize@skip@two
474   \@bmpsize@skip@one
475   \ifnum\pdf@strcmp{\bmpsize@temp}{FFD8FF}=\z@
476   \else
477     \expandafter\@bmpsize@stop
478   \fi
479   \def\bmpsize@offset{2}%
480   \def\bmpsize@exifdensity{0}%
481   \@bmpsize@loop{%
482     \@bmpsize@read{#1}{4}{\bmpsize@offset}%
483     \@bmpsize@grab\bmpsize@temp{1}%
484     \@bmpsize@skip@one
485     \ifnum\pdf@strcmp{\bmpsize@temp}{FF}=\z@
486     \else
487       \expandafter\@bmpsize@stop
488     \fi
489     \@bmpsize@num@one\bmpsize@temp
490     \ifnum\bmpsize@temp=218\relax
491       \expandafter\@firstofone
492     \else
493       \expandafter\@gobble
494     \fi
495     {%
496       \@bmpsize@stop
497     }%
498     \ifnum\bmpsize@temp=224\relax
499       \expandafter\@firstoftwo
500     \else
501       \expandafter\@secondoftwo
502     \fi
503     {%
504       \@bmpsize@num@two\bmpsize@length
505       \ifnum\bmpsize@exifdensity=0\relax
506         \expandafter\@firstofone
507       \else
508         \expandafter\@gobble
509       \fi
510       {%
511         \unless\ifnum\bmpsize@length<16\relax
512           \expandafter\@firstofone
513         \else
514           \expandafter\@gobble
515         \fi
516         {%
517           \@bmpsize@read{#1}{12}{\numexpr\bmpsize@offset+4\relax}%
518           \@bmpsize@grab\bmpsize@temp{5}%
519           \@bmpsize@skip@four
520           \@bmpsize@skip@one
521           \ifnum\pdf@strcmp{\bmpsize@temp}{4A46494600}=\z@
522             \expandafter\@firstofone
523           \else
524             \expandafter\@gobble
525           \fi
526           {%
```

```
527        \ifnum\bmpsize@length<16\relax
528          \expandafter\@bmpsize@stop
529        \fi
530        \@bmpsize@skip@two
531        \@bmpsize@num@one\bmpsize@temp
532        \ifnum\bmpsize@temp=1\relax
533          \expandafter\@firstoftwo
534        \else
535          \expandafter\@secondoftwo
536        \fi
537        {%
538          \def\bmpsize@unit{72.27pt}%
539        }{%
540          \ifnum\bmpsize@temp=2\relax
541            \expandafter\@firstofone
542          \else
543            \expandafter\@gobble
544          \fi
545          {%
546            \def\bmpsize@unit{1cm}%
547          }%
548        }%
549        \@bmpsize@num@two\bmpsize@pixelx
550        \@bmpsize@num@two\bmpsize@pixely
551      }%
552    }%
553    }%
554  }{%
555    \ifnum\bmpsize@temp=225\relax
556      \expandafter\@firstoftwo
557    \else
558      \expandafter\@secondoftwo
559    \fi
560    {%
561      \@bmpsize@num@two\bmpsize@length
562      \unless\ifnum\bmpsize@length<20\relax
563        \expandafter\@firstofone
564      \else
565        \expandafter\@gobble
566      \fi
567      {%
568        \@bmpsize@read{#1}{20}{\numexpr\bmpsize@offset+4\relax}%
569        \@bmpsize@grab\bmpsize@temp{6}%
570        \@bmpsize@skip@four
571        \@bmpsize@skip@two
572        \ifnum\pdf@strcmp{\bmpsize@temp}{457869660000}=\z@
573          \expandafter\@firstofone
574        \else
575          \expandafter\@gobble
576        \fi
577        {%
578          \edef\bmpsize@exifoffset{\the\numexpr\bmpsize@offset+10}%
579          \@bmpsize@grab\bmpsize@temp{2}%
580          \@bmpsize@skip@two
581          \ifnum\pdf@strcmp{\bmpsize@temp}{4949}=\z@
582            \expandafter\@firstoftwo
583          \else
584            \expandafter\@secondoftwo
585          \fi
586          {%
587            \@bmpsize@bigendianfalse
588          }{%
```

```
589          \ifnum\pdf@strcmp{\bmpsize@temp}{4D4D}=\z@
590          \else
591            \expandafter\@bmpsize@stop
592          \fi
593        }%
594        \@bmpsize@num@two\bmpsize@temp
595        \ifnum\bmpsize@temp=42\relax
596        \else
597          \expandafter\@bmpsize@stop
598        \fi
599        \@bmpsize@num@four\bmpsize@temp
600        \ifnum\bmpsize@temp>0\relax
601        \else
602          \expandafter\@bmpsize@stop
603        \fi
604      \edef\bmpsize@off{\the\numexpr\bmpsize@temp+\bmpsize@exifoffset}%
605      \@bmpsize@read{#1}{2}{\bmpsize@off}%
606      \@bmpsize@num@two\bmpsize@entries
607      \edef\bmpsize@off{\the\numexpr\bmpsize@off+2}%
608      \@bmpsize@loop{%
609        \ifnum\bmpsize@entries=0\relax
610          \expandafter\@firstofone
611        \else
612          \expandafter\@gobble
613        \fi
614        {%
615          \@bmpsize@break
616        }%
617        \edef\bmpsize@entries{\the\numexpr\bmpsize@entries-1}%
618        \@bmpsize@read{#1}{12}{\bmpsize@off}%
619        \edef\bmpsize@off{\the\numexpr\bmpsize@off+12}%
620        \@bmpsize@num@two\bmpsize@tag
621        \ifnum\bmpsize@tag=296\relax
622          \expandafter\@firstofone
623        \else
624          \expandafter\@gobble
625        \fi
626        {%
627          \@bmpsize@skip@four
628          \@bmpsize@skip@two
629          \@bmpsize@num@two\bmpsize@temp
630          \ifcase\bmpsize@temp\relax
631          \or
632            \let\bmpsize@unit\relax
633          \or
634            \def\bmpsize@unit{72.27pt}%
635          \or
636            \def\bmpsize@unit{1cm}%
637          \else
638            \let\bmpsize@unit\relax
639          \fi
640          \ifcase\bmpsize@temp\relax
641          \or
642          \or
643            \def\bmpsize@exifdensity{1}%
644          \or
645            \def\bmpsize@exifdensity{1}%
646          \else
647            \let\bmpsize@exifdensity\bmpsize@exifdensity
648          \fi
649        }%
650        \ifnum\bmpsize@tag=274\relax
```

18

```
651          \expandafter\@firstofone
652        \else
653          \expandafter\@gobble
654        \fi
655        {%
656          \@bmpsize@skip@four
657          \@bmpsize@skip@two
658          \@bmpsize@num@two\bmpsize@temp
659          \unless\ifnum\bmpsize@temp<0\relax
660            \expandafter\@firstofone
661          \else
662            \expandafter\@gobble
663          \fi
664          {%
665            \unless\ifnum\bmpsize@temp>8\relax
666              \expandafter\@firstofone
667            \else
668              \expandafter\@gobble
669            \fi
670            {%
671              \let\bmpsize@orientation\bmpsize@temp
672            }%
673          }%
674        }%
675        \ifnum\bmpsize@tag=282\relax
676          \expandafter\@firstofone
677        \else
678          \expandafter\@gobble
679        \fi
680        {%
681          \@bmpsize@skip@four
682          \@bmpsize@skip@two
683          \@bmpsize@num@four\bmpsize@temp
684          \@bmpsize@read{#1}{8}{\numexpr\bmpsize@temp+\bmpsize@exi-
     foffset\relax}%
685          \@bmpsize@num@four\bmpsize@pixelx
686          \@bmpsize@num@four\bmpsize@temp
687          \ifnum\bmpsize@temp=1\relax
688            \expandafter\@gobble
689          \else
690            \expandafter\@firstofone
691          \fi
692          {%
693            \edef\bmpsize@pixelxdenom{\the\numexpr\bmpsize@temp}%
694          }%
695        }%
696        \ifnum\bmpsize@tag=283\relax
697          \expandafter\@firstofone
698        \else
699          \expandafter\@gobble
700        \fi
701        {%
702          \@bmpsize@skip@four
703          \@bmpsize@skip@two
704          \@bmpsize@num@four\bmpsize@temp
705          \@bmpsize@read{#1}{8}{\numexpr\bmpsize@temp+\bmpsize@exi-
     foffset\relax}%
706          \@bmpsize@num@four\bmpsize@pixely
707          \@bmpsize@num@four\bmpsize@temp
708          \ifnum\bmpsize@temp=1\relax
709            \expandafter\@gobble
710          \else
```

```
711          \expandafter\@firstofone
712        \fi
713        {%
714          \edef\bmpsize@pixelydenom{\the\numexpr\bmpsize@temp}%
715        }%
716      }%
717      }%
718      \@bmpsize@bigendiantrue
719    }%
720    }%
721    }{%
722    \edef\bmpsize@temp{\the\numexpr\bmpsize@temp-192}%
723    \ifcase\bmpsize@temp\relax
724    \or
725    \or
726    \or
727    \or
728      \def\bmpsize@temp{-1}%
729    \or
730    \or
731    \or
732    \or
733      \def\bmpsize@temp{-1}%
734    \or
735    \or
736    \or
737    \or
738      \def\bmpsize@temp{-1}%
739    \or
740    \or
741    \or
742    \else
743      \def\bmpsize@temp{-1}%
744    \fi
745    \ifnum\bmpsize@temp=-1\relax
746      \expandafter\@gobble
747    \else
748      \expandafter\@firstofone
749    \fi
750    {%
751    \@bmpsize@read{#1}{4}{\numexpr\bmpsize@offset+5\relax}%
752    \@bmpsize@num@two\bmpsize@pixelheight
753    \@bmpsize@num@two\bmpsize@pixelwidth
754    \ifnum\bmpsize@pixelheight=0\relax
755      \expandafter\@firstofone
756    \else
757      \expandafter\@gobble
758    \fi
759    {%
760      \let\bmpsize@pixelheight\relax
761      \@bmpsize@stop
762    }%
763    \@bmpsize@ok
764    \@bmpsize@stop
765    }%
766    \@bmpsize@num@two\bmpsize@length
767    }%
768    }%
769    \edef\bmpsize@offset{\the\numexpr\bmpsize@offset+\bmpsize@length+2}%
770  }%
771  \@bmpsize@stop
772  \@nil
```

```
773    \@bmpsize@end
774 }%
775 ⟨/base⟩
```

2.2.3 bmp

```
begin bmp
little-endian

read 26 0
grab 2 -> $temp
check streq $temp ["BM"]
skip 12
% header size is 4 bytes in V3+, unknown for V1, V2,
% known header sizes fit in 2 bytes
num 2  -> $temp
if numeq $temp 12 % V1
  skip 2
  num 2 -> $pixelwidth
  num 2 -> $pixelheight
  % no resolution entries
  ok
  stop
fi
if numeq $temp 64 % V2
  skip 2
  num 2 -> $pixelwidth
  num 2 -> $pixelheight
  % missing specification for resolution
  ok
  stop
fi
% V3, V4, V5
skip 2
num 4 -> $pixelwidth
absnum 4 -> $pixelheight
ok
read 8 38
num 4 -> $pixelx
num 4 -> $pixely
assign {100cm} -> $unit
end
```

\bmpsize@read@bmp

```
776 ⟨*base⟩
777 \def\bmpsize@read@bmp#1{%
778   \@bmpsize@init
779   \@bmpsize@bigendianfalse
780   \@bmpsize@read{#1}{26}{0}%
781   \@bmpsize@grab\bmpsize@temp{2}%
782   \@bmpsize@skip@two
783   \ifnum\pdf@strcmp{\bmpsize@temp}{424D}=\z@
784   \else
785     \expandafter\@bmpsize@stop
786   \fi
787   \@bmpsize@skip@four
788   \@bmpsize@skip@four
789   \@bmpsize@skip@four
790   \@bmpsize@num@two\bmpsize@temp
791   \ifnum\bmpsize@temp=12\relax
792     \expandafter\@firstofone
793   \else
794     \expandafter\@gobble
```

```
795   \fi
796   {%
797     \@bmpsize@skip@two
798     \@bmpsize@num@two\bmpsize@pixelwidth
799     \@bmpsize@num@two\bmpsize@pixelheight
800     \@bmpsize@ok
801     \@bmpsize@stop
802   }%
803   \ifnum\bmpsize@temp=64\relax
804     \expandafter\@firstofone
805   \else
806     \expandafter\@gobble
807   \fi
808   {%
809     \@bmpsize@skip@two
810     \@bmpsize@num@two\bmpsize@pixelwidth
811     \@bmpsize@num@two\bmpsize@pixelheight
812     \@bmpsize@ok
813     \@bmpsize@stop
814   }%
815   \@bmpsize@skip@two
816   \@bmpsize@num@four\bmpsize@pixelwidth
817   \@bmpsize@absnumtrue
818   \@bmpsize@num@four\bmpsize@pixelheight
819   \@bmpsize@absnumfalse
820   \@bmpsize@ok
821   \@bmpsize@read{#1}{8}{38}%
822   \@bmpsize@num@four\bmpsize@pixelx
823   \@bmpsize@num@four\bmpsize@pixely
824   \def\bmpsize@unit{100cm}%
825   \@bmpsize@stop
826   \@nil
827   \@bmpsize@end
828 }%
829 ⟨/base⟩
```

2.2.4 gif

```
begin gif
little-endian

% Header
read 13 0
grab 3      -> $temp
check streq $temp ["GIF"]
skip 3      % version

% Logical Screen Descriptor
num 2      -> $pixelwidth
num 2      -> $pixelheight
skip 2
num 1      -> $temp % Pixel Aspect Ratio
if numeq $temp 0
else
  assign numexpr($temp + 15) -> $pixelx
  assign {64}     -> $pixely
fi
ok
end
```

\bmpsize@read@gif

```
830 ⟨*base⟩
831 \def\bmpsize@read@gif#1{%
```

22

```
832  \@bmpsize@init
833  \@bmpsize@bigendianfalse
834  \@bmpsize@read{#1}{13}{0}%
835  \@bmpsize@grab\bmpsize@temp{3}%
836  \@bmpsize@skip@two
837  \@bmpsize@skip@one
838  \ifnum\pdf@strcmp{\bmpsize@temp}{474946}=\z@
839  \else
840    \expandafter\@bmpsize@stop
841  \fi
842  \@bmpsize@skip@two
843  \@bmpsize@skip@one
844  \@bmpsize@num@two\bmpsize@pixelwidth
845  \@bmpsize@num@two\bmpsize@pixelheight
846  \@bmpsize@skip@two
847  \@bmpsize@num@one\bmpsize@temp
848  \ifnum\bmpsize@temp=0\relax
849    \expandafter\@gobble
850  \else
851    \expandafter\@firstofone
852  \fi
853  {%
854    \edef\bmpsize@pixelx{\the\numexpr\bmpsize@temp+15}%
855    \def\bmpsize@pixely{64}%
856  }%
857  \@bmpsize@ok
858  \@bmpsize@stop
859  \@nil
860  \@bmpsize@end
861 }%
862 ⟨/base⟩
```

2.2.5   tiff

```
begin tiff
% defaults
assign {72.27pt} -> $unit

% Image File Header
read 8 0
grab 2 -> $temp
if streq $temp ["II"]
  little-endian
else
  check streq $temp ["MM"]
  big-endian
fi
num 2 -> $temp
check numeq $temp 42
num 4 -> $offset % first IFD (Image File Directory)

% First IFD
read 2 $offset
assign numexpr($offset + 2) -> $offset
num 2 -> $entries
ok % must rely on checks at the end
loop
  if numeq $entries 0
    stop
  fi
  assign numexpr($entries - 1) -> $entries
  % entry format:
  % 2 tag
```

23

```
                 % 2 field type
                 % 4 count
                 % 4 value/offset
                 read 12 $offset
                 assign numexpr($offset + 12) -> $offset
                 num 2 -> $tag % tag
                 if numeq $temp 296 % ResolutionUnit
                   skip 6 % type: 3 (short), count: 1
                   num 2 -> $temp
                   ifcase $temp
                   or % 1
                     clear $unit
                   or % 2
                     assign {72.27pt} -> $unit
                   or % 3
                     assign {1cm} -> $unit
                   else
                     clear $unit
                   fi
                 fi
                 if numeq $tag 256 % ImageWidth
                   skip 6
                   num 4 -> $pixelwidth
                 fi
                 if numeq $tag 257 % ImageLength
                   skip 6
                   num 4 -> $pixelheight
                 fi
                 if numeq $tag 282 % XResolution
                   skip 6
                   num 4 -> $temp
                   read 8 $temp
                   num 4 -> $pixelx
                   num 4 -> $temp
                   if numeq $temp 1
                   else
                     assign numexpr($temp) -> $pixelxdenom
                     % div $pixelx $temp -> $pixelx
                   fi
                 fi
                 if numeq $tag 283 % YResolution
                   skip 6
                   num 4 -> $temp
                   read 8 $temp
                   num 4 -> $pixely
                   num 4 -> $temp
                   if numeq $temp 1
                   else
                     assign numexpr($temp) -> $pixelydenom
                     % div $pixely $temp -> $pixely
                   fi
                 fi
               repeat
               end
```

\bmpsize@read@tiff

```
863 ⟨*base⟩
864 \def\bmpsize@read@tiff#1{%
865   \@bmpsize@init
866   \def\bmpsize@unit{72.27pt}%
867   \@bmpsize@read{#1}{8}{0}%
868   \@bmpsize@grab\bmpsize@temp{2}%
869   \@bmpsize@skip@two
```

24

```
870  \ifnum\pdf@strcmp{\bmpsize@temp}{4949}=\z@
871    \expandafter\@firstoftwo
872  \else
873    \expandafter\@secondoftwo
874  \fi
875  {%
876    \@bmpsize@bigendianfalse
877  }{%
878    \ifnum\pdf@strcmp{\bmpsize@temp}{4D4D}=\z@
879    \else
880      \expandafter\@bmpsize@stop
881    \fi
882    \@bmpsize@bigendiantrue
883  }%
884  \@bmpsize@num@two\bmpsize@temp
885  \ifnum\bmpsize@temp=42\relax
886  \else
887    \expandafter\@bmpsize@stop
888  \fi
889  \@bmpsize@num@four\bmpsize@offset
890  \@bmpsize@read{#1}{2}{\bmpsize@offset}%
891  \edef\bmpsize@offset{\the\numexpr\bmpsize@offset+2}%
892  \@bmpsize@num@two\bmpsize@entries
893  \@bmpsize@ok
894  \@bmpsize@loop{%
895    \ifnum\bmpsize@entries=0\relax
896      \expandafter\@firstofone
897    \else
898      \expandafter\@gobble
899    \fi
900    {%
901      \@bmpsize@stop
902    }%
903    \edef\bmpsize@entries{\the\numexpr\bmpsize@entries-1}%
904    \@bmpsize@read{#1}{12}{\bmpsize@offset}%
905    \edef\bmpsize@offset{\the\numexpr\bmpsize@offset+12}%
906    \@bmpsize@num@two\bmpsize@tag
907    \ifnum\bmpsize@temp=296\relax
908      \expandafter\@firstofone
909    \else
910      \expandafter\@gobble
911    \fi
912    {%
913      \@bmpsize@skip@four
914      \@bmpsize@skip@two
915      \@bmpsize@num@two\bmpsize@temp
916      \ifcase\bmpsize@temp\relax
917      \or
918        \let\bmpsize@unit\relax
919      \or
920        \def\bmpsize@unit{72.27pt}%
921      \or
922        \def\bmpsize@unit{1cm}%
923      \else
924        \let\bmpsize@unit\relax
925      \fi
926    }%
927    \ifnum\bmpsize@tag=256\relax
928      \expandafter\@firstofone
929    \else
930      \expandafter\@gobble
931    \fi
```

```
932    {%
933      \@bmpsize@skip@four
934      \@bmpsize@skip@two
935      \@bmpsize@num@four\bmpsize@pixelwidth
936    }%
937    \ifnum\bmpsize@tag=257\relax
938      \expandafter\@firstofone
939    \else
940      \expandafter\@gobble
941    \fi
942    {%
943      \@bmpsize@skip@four
944      \@bmpsize@skip@two
945      \@bmpsize@num@four\bmpsize@pixelheight
946    }%
947    \ifnum\bmpsize@tag=282\relax
948      \expandafter\@firstofone
949    \else
950      \expandafter\@gobble
951    \fi
952    {%
953      \@bmpsize@skip@four
954      \@bmpsize@skip@two
955      \@bmpsize@num@four\bmpsize@temp
956      \@bmpsize@read{#1}{8}{\bmpsize@temp}%
957      \@bmpsize@num@four\bmpsize@pixelx
958      \@bmpsize@num@four\bmpsize@temp
959      \ifnum\bmpsize@temp=1\relax
960        \expandafter\@gobble
961      \else
962        \expandafter\@firstofone
963      \fi
964      {%
965        \edef\bmpsize@pixelxdenom{\the\numexpr\bmpsize@temp}%
966      }%
967    }%
968    \ifnum\bmpsize@tag=283\relax
969      \expandafter\@firstofone
970    \else
971      \expandafter\@gobble
972    \fi
973    {%
974      \@bmpsize@skip@four
975      \@bmpsize@skip@two
976      \@bmpsize@num@four\bmpsize@temp
977      \@bmpsize@read{#1}{8}{\bmpsize@temp}%
978      \@bmpsize@num@four\bmpsize@pixely
979      \@bmpsize@num@four\bmpsize@temp
980      \ifnum\bmpsize@temp=1\relax
981        \expandafter\@gobble
982      \else
983        \expandafter\@firstofone
984      \fi
985      {%
986        \edef\bmpsize@pixelydenom{\the\numexpr\bmpsize@temp}%
987      }%
988    }%
989    }%
990    \@bmpsize@stop
991    \@nil
992    \@bmpsize@end
993 }%
```

2.2.6   pnm

```
begin pnm
assign {0} -> $offset
read 3 $offset
assign {3} -> $offset
grab 1 -> $temp
check streq $temp ["P"]
grab 1 -> $temp
check strge $temp ["1"]
check strle $temp ["6"]
% ensure one white space
grab 1 -> $temp
if iswhite $temp
else
  stop
fi
loop
  % skip white space
  fillbuf
  grab 1 -> $temp
  if iswhite $temp
  else
    if streq $temp ["#"]
      % ignore comments
      loop
        fillbuf
        grab 1 -> $temp
        if streq $temp [0x0A]
          break
        else
          if streq $temp [0x0D]
            break
          fi
        fi
      repeat
    else
      pushback $temp
      break
    fi
  fi
repeat
assign {} -> $tempnum
loop
  fillbuf
  grab 1 -> $temp
  if isdigit $temp
    append $tempnum $temp -> $tempnum
  else
    if iswhite $temp
      break
    else
      stop
    fi
  fi
repeat
assign unescapehex($tempnum) -> $pixelwidth
loop
  fillbuf
  grab 1 -> $temp
  if iswhite $temp
```

```
        else
          pushback $temp
          break
        fi
      repeat
      assign {} -> $tempnum
      loop
        fillbuf
        grab 1 -> $temp
        if isdigit $temp
          append $tempnum $temp -> $tempnum
        else
          if iswhite $temp
            break
          else
            stop
          fi
        fi
      repeat
      assign unescapehex($tempnum) -> $pixelheight
      ok
      end
```

\bmpsize@read@pnm

```
 995 ⟨*base⟩
 996 \def\bmpsize@read@pnm#1{%
 997   \@bmpsize@init
 998   \def\bmpsize@offset{0}%
 999   \@bmpsize@read{#1}{3}{\bmpsize@offset}%
1000   \def\bmpsize@offset{3}%
1001   \@bmpsize@grab\bmpsize@temp{1}%
1002   \@bmpsize@skip@one
1003   \ifnum\pdf@strcmp{\bmpsize@temp}{50}=\z@
1004   \else
1005     \expandafter\@bmpsize@stop
1006   \fi
1007   \@bmpsize@grab\bmpsize@temp{1}%
1008   \@bmpsize@skip@one
1009   \ifnum\pdf@strcmp{\bmpsize@temp}{31}<\z@
1010     \expandafter\@bmpsize@stop
1011   \fi
1012   \ifnum\pdf@strcmp{\bmpsize@temp}{36}>\z@
1013     \expandafter\@bmpsize@stop
1014   \fi
1015   \@bmpsize@grab\bmpsize@temp{1}%
1016   \@bmpsize@skip@one
1017   \ifcase 0\@bmpsize@iswhite\bmpsize@temp
1018     \expandafter\@gobble
1019   \else
1020     \expandafter\@firstofone
1021   \fi
1022   {%
1023     \@bmpsize@stop
1024   }%
1025   \@bmpsize@loop{%
1026     \@bmpsize@fillbuf{#1}%
1027     \@bmpsize@grab\bmpsize@temp{1}%
1028     \@bmpsize@skip@one
1029     \ifcase 0\@bmpsize@iswhite\bmpsize@temp
1030       \expandafter\@gobble
1031     \else
1032       \expandafter\@firstofone
1033     \fi
```

```latex
1034    {%
1035      \ifnum\pdf@strcmp{\bmpsize@temp}{23}=\z@
1036        \expandafter\@firstoftwo
1037      \else
1038        \expandafter\@secondoftwo
1039      \fi
1040      {%
1041        \@bmpsize@loop{%
1042          \@bmpsize@fillbuf{#1}%
1043          \@bmpsize@grab\bmpsize@temp{1}%
1044          \@bmpsize@skip@one
1045          \ifnum\pdf@strcmp{\bmpsize@temp}{0A}=\z@
1046            \expandafter\@firstoftwo
1047          \else
1048            \expandafter\@secondoftwo
1049          \fi
1050          {%
1051            \@bmpsize@break
1052          }{%
1053            \ifnum\pdf@strcmp{\bmpsize@temp}{0D}=\z@
1054              \expandafter\@firstofone
1055            \else
1056              \expandafter\@gobble
1057            \fi
1058            {%
1059              \@bmpsize@break
1060            }%
1061          }%
1062        }%
1063      }{%
1064        \@bmpsize@pushback\bmpsize@temp
1065        \@bmpsize@break
1066      }%
1067    }%
1068  }%
1069  \def\bmpsize@tempnum{}%
1070  \@bmpsize@loop{%
1071    \@bmpsize@fillbuf{#1}%
1072    \@bmpsize@grab\bmpsize@temp{1}%
1073    \@bmpsize@skip@one
1074    \ifcase 0\@bmpsize@isdigit\bmpsize@temp
1075      \expandafter\@firstoftwo
1076    \else
1077      \expandafter\@secondoftwo
1078    \fi
1079    {%
1080      \@bmpsize@append\bmpsize@tempnum\bmpsize@tempnum\bmpsize@temp
1081    }{%
1082      \ifcase 0\@bmpsize@iswhite\bmpsize@temp
1083        \expandafter\@firstoftwo
1084      \else
1085        \expandafter\@secondoftwo
1086      \fi
1087      {%
1088        \@bmpsize@break
1089      }{%
1090        \@bmpsize@stop
1091      }%
1092    }%
1093  }%
1094  \edef\bmpsize@pixelwidth{\pdf@unescapehex{\bmpsize@tempnum}}%
1095  \@bmpsize@loop{%
```

```
1096    \@bmpsize@fillbuf{#1}%
1097    \@bmpsize@grab\bmpsize@temp{1}%
1098    \@bmpsize@skip@one
1099    \ifcase 0\@bmpsize@iswhite\bmpsize@temp
1100      \expandafter\@gobble
1101    \else
1102      \expandafter\@firstofone
1103    \fi
1104    {%
1105      \@bmpsize@pushback\bmpsize@temp
1106      \@bmpsize@break
1107    }%
1108  }%
1109  \def\bmpsize@tempnum{}%
1110  \@bmpsize@loop{%
1111    \@bmpsize@fillbuf{#1}%
1112    \@bmpsize@grab\bmpsize@temp{1}%
1113    \@bmpsize@skip@one
1114    \ifcase 0\@bmpsize@isdigit\bmpsize@temp
1115      \expandafter\@firstoftwo
1116    \else
1117      \expandafter\@secondoftwo
1118    \fi
1119    {%
1120      \@bmpsize@append\bmpsize@tempnum\bmpsize@tempnum\bmpsize@temp
1121    }{%
1122      \ifcase 0\@bmpsize@iswhite\bmpsize@temp
1123        \expandafter\@firstoftwo
1124      \else
1125        \expandafter\@secondoftwo
1126      \fi
1127      {%
1128        \@bmpsize@break
1129      }{%
1130        \@bmpsize@stop
1131      }%
1132    }%
1133  }%
1134  \edef\bmpsize@pixelheight{\pdf@unescapehex{\bmpsize@tempnum}}%
1135  \@bmpsize@ok
1136  \@bmpsize@stop
1137  \@nil
1138  \@bmpsize@end
1139 }%
1140 ⟨/base⟩
```

2.2.7 pam

```
begin pam
read 3 0
assign {3} -> $offset
assign $offset -> $off
grab 3 -> $temp
check streq $temp ["P7" 0x0A]
loop
  fillbuf
  grab 1 -> $temp
  if iswhite $temp
    % ignore white space
    assign numexpr($off + 1) -> $off
  else
    if streq $temp ["#"]
      % ignore comment line
```

```
    assign numexpr($off + 1) -> $off
  loop
    fillbuf
    grab 1 -> $temp
    assign numexpr($off + 1) -> $off
    if streq $temp [0x0A]
      break
    fi
  repeat
else
  read 6 $off
  assign numexpr($off + 6) -> $offset
  grab 5 -> $head
  if streq $head ["WIDTH"]
    assign numexpr($off + 5) -> $off
    % skip white space
    loop
      fillbuf
      grab 1 -> $temp
      if iswhite $temp
        assign numexpr($off + 1) -> $off
      else
        if isdigit $temp
          assign numexpr($off + 1) -> $off
          break
        else
          % error
          stop
        fi
      fi
    repeat
    % read number
    assign $temp -> $tempnum
    loop
      fillbuf
      grab 1 -> $temp
      if isdigit $temp
        assign numexpr($off + 1) -> $off
        append $tempnum $temp -> $tempnum
      else
        pushback $temp
        break
      fi
    repeat
    % skip to end of line
    loop
      fillbuf
      grab 1 -> $temp
      assign numexpr($off + 1) -> $off
      if streq $temp [0x0A]
        break
      fi
    repeat
    assign unescapehex($tempnum) -> $pixelwidth
  else
    grab 1 -> $temp
    append $head $temp -> $head
    if streq $head ["ENDHDR"]
      % last header line
      ok
      stop
    else
```

```
            if streq $head ["HEIGHT"]
              assign numexpr($off + 6) -> $off
              % skip white space
              loop
                fillbuf
                grab 1 -> $temp
                if iswhite $temp
                  assign numexpr($off + 1) -> $off
                else
                  if isdigit $temp
                    assign numexpr($off + 1) -> $off
                    break
                  else
                    % error
                    stop
                  fi
                fi
              repeat
              % read number
              assign $temp -> $tempnum
              loop
                fillbuf
                grab 1 -> $temp
                if isdigit $temp
                  assign numexpr($off + 1) -> $off
                  append $tempnum $temp -> $tempnum
                else
                  pushback $temp
                  break
                fi
              repeat
              % skip to end of line
              loop
                fillbuf
                grab 1 -> $temp
                assign numexpr($off + 1) -> $off
                if streq $temp [0x0A]
                  break
                fi
              repeat
              assign unescapehex($tempnum) -> $pixelheight
            else
              % ignore unknown header line
              pushback $head
              loop
                fillbuf
                grab 1 -> $temp
                assign numexpr($off + 1) -> $off
                if streq $temp [0x0A]
                  break
                fi
              repeat
            fi
          fi
        fi
      fi
    fi
  repeat
end
```

\bmpsize@read@pam

1141 ⟨*base⟩
1142 \def\bmpsize@read@pam#1{%

```
1143  \@bmpsize@init
1144  \@bmpsize@read{#1}{3}{0}%
1145  \def\bmpsize@offset{3}%
1146  \let\bmpsize@off\bmpsize@offset
1147  \@bmpsize@grab\bmpsize@temp{3}%
1148  \@bmpsize@skip@two
1149  \@bmpsize@skip@one
1150  \ifnum\pdf@strcmp{\bmpsize@temp}{50370A}=\z@
1151  \else
1152    \expandafter\@bmpsize@stop
1153  \fi
1154  \@bmpsize@loop{%
1155    \@bmpsize@fillbuf{#1}%
1156    \@bmpsize@grab\bmpsize@temp{1}%
1157    \@bmpsize@skip@one
1158    \ifcase 0\@bmpsize@iswhite\bmpsize@temp
1159      \expandafter\@firstoftwo
1160    \else
1161      \expandafter\@secondoftwo
1162    \fi
1163    {%
1164      \edef\bmpsize@off{\the\numexpr\bmpsize@off+1}%
1165    }{%
1166      \ifnum\pdf@strcmp{\bmpsize@temp}{23}=\z@
1167        \expandafter\@firstoftwo
1168      \else
1169        \expandafter\@secondoftwo
1170      \fi
1171      {%
1172        \edef\bmpsize@off{\the\numexpr\bmpsize@off+1}%
1173        \@bmpsize@loop{%
1174          \@bmpsize@fillbuf{#1}%
1175          \@bmpsize@grab\bmpsize@temp{1}%
1176          \@bmpsize@skip@one
1177          \edef\bmpsize@off{\the\numexpr\bmpsize@off+1}%
1178          \ifnum\pdf@strcmp{\bmpsize@temp}{0A}=\z@
1179            \expandafter\@firstofone
1180          \else
1181            \expandafter\@gobble
1182          \fi
1183          {%
1184            \@bmpsize@break
1185          }%
1186        }%
1187      }{%
1188        \@bmpsize@read{#1}{6}{\bmpsize@off}%
1189        \edef\bmpsize@offset{\the\numexpr\bmpsize@off+6}%
1190        \@bmpsize@grab\bmpsize@head{5}%
1191        \@bmpsize@skip@four
1192        \@bmpsize@skip@one
1193        \ifnum\pdf@strcmp{\bmpsize@head}{5749445448}=\z@
1194          \expandafter\@firstoftwo
1195        \else
1196          \expandafter\@secondoftwo
1197        \fi
1198        {%
1199          \edef\bmpsize@off{\the\numexpr\bmpsize@off+5}%
1200          \@bmpsize@loop{%
1201            \@bmpsize@fillbuf{#1}%
1202            \@bmpsize@grab\bmpsize@temp{1}%
1203            \@bmpsize@skip@one
1204            \ifcase 0\@bmpsize@iswhite\bmpsize@temp
```

33

```
1205          \expandafter\@firstoftwo
1206        \else
1207          \expandafter\@secondoftwo
1208        \fi
1209        {%
1210          \edef\bmpsize@off{\the\numexpr\bmpsize@off+1}%
1211        }{%
1212         \ifcase 0\@bmpsize@isdigit\bmpsize@temp
1213           \expandafter\@firstoftwo
1214         \else
1215           \expandafter\@secondoftwo
1216         \fi
1217         {%
1218           \edef\bmpsize@off{\the\numexpr\bmpsize@off+1}%
1219           \@bmpsize@break
1220         }{%
1221           \@bmpsize@stop
1222         }%
1223        }%
1224      }%
1225      \let\bmpsize@tempnum\bmpsize@temp
1226      \@bmpsize@loop{%
1227        \@bmpsize@fillbuf{#1}%
1228        \@bmpsize@grab\bmpsize@temp{1}%
1229        \@bmpsize@skip@one
1230        \ifcase 0\@bmpsize@isdigit\bmpsize@temp
1231          \expandafter\@firstoftwo
1232        \else
1233          \expandafter\@secondoftwo
1234        \fi
1235        {%
1236          \edef\bmpsize@off{\the\numexpr\bmpsize@off+1}%
1237              \@bmpsize@append\bmpsize@tempnum\bmpsize@tempnum\bmp-
      size@temp
1238        }{%
1239          \@bmpsize@pushback\bmpsize@temp
1240          \@bmpsize@break
1241        }%
1242      }%
1243      \@bmpsize@loop{%
1244        \@bmpsize@fillbuf{#1}%
1245        \@bmpsize@grab\bmpsize@temp{1}%
1246        \@bmpsize@skip@one
1247        \edef\bmpsize@off{\the\numexpr\bmpsize@off+1}%
1248        \ifnum\pdf@strcmp{\bmpsize@temp}{0A}=\z@
1249          \expandafter\@firstofone
1250        \else
1251          \expandafter\@gobble
1252        \fi
1253        {%
1254          \@bmpsize@break
1255        }%
1256      }%
1257      \edef\bmpsize@pixelwidth{\pdf@unescapehex{\bmpsize@tempnum}}%
1258    }{%
1259      \@bmpsize@grab\bmpsize@temp{1}%
1260      \@bmpsize@skip@one
1261      \@bmpsize@append\bmpsize@head\bmpsize@head\bmpsize@temp
1262      \ifnum\pdf@strcmp{\bmpsize@head}{454E44484452}=\z@
1263      \expandafter\@firstoftwo
1264    \else
1265      \expandafter\@secondoftwo
```

```
1266        \fi
1267        {%
1268         \@bmpsize@ok
1269         \@bmpsize@stop
1270        }{%
1271         \ifnum\pdf@strcmp{\bmpsize@head}{484549474854}=\z@
1272           \expandafter\@firstoftwo
1273         \else
1274           \expandafter\@secondoftwo
1275         \fi
1276         {%
1277          \edef\bmpsize@off{\the\numexpr\bmpsize@off+6}%
1278          \@bmpsize@loop{%
1279           \@bmpsize@fillbuf{#1}%
1280           \@bmpsize@grab\bmpsize@temp{1}%
1281           \@bmpsize@skip@one
1282           \ifcase 0\@bmpsize@iswhite\bmpsize@temp
1283            \expandafter\@firstoftwo
1284           \else
1285            \expandafter\@secondoftwo
1286           \fi
1287           {%
1288            \edef\bmpsize@off{\the\numexpr\bmpsize@off+1}%
1289           }{%
1290            \ifcase 0\@bmpsize@isdigit\bmpsize@temp
1291             \expandafter\@firstoftwo
1292            \else
1293             \expandafter\@secondoftwo
1294            \fi
1295            {%
1296             \edef\bmpsize@off{\the\numexpr\bmpsize@off+1}%
1297             \@bmpsize@break
1298            }{%
1299             \@bmpsize@stop
1300            }%
1301           }%
1302          }%
1303          \let\bmpsize@tempnum\bmpsize@temp
1304          \@bmpsize@loop{%
1305           \@bmpsize@fillbuf{#1}%
1306           \@bmpsize@grab\bmpsize@temp{1}%
1307           \@bmpsize@skip@one
1308           \ifcase 0\@bmpsize@isdigit\bmpsize@temp
1309            \expandafter\@firstoftwo
1310           \else
1311            \expandafter\@secondoftwo
1312           \fi
1313           {%
1314            \edef\bmpsize@off{\the\numexpr\bmpsize@off+1}%
1315             \@bmpsize@append\bmpsize@tempnum\bmpsize@tempnum\bmp-
      size@temp
1316           }{%
1317            \@bmpsize@pushback\bmpsize@temp
1318            \@bmpsize@break
1319           }%
1320          }%
1321          \@bmpsize@loop{%
1322           \@bmpsize@fillbuf{#1}%
1323           \@bmpsize@grab\bmpsize@temp{1}%
1324           \@bmpsize@skip@one
1325           \edef\bmpsize@off{\the\numexpr\bmpsize@off+1}%
1326           \ifnum\pdf@strcmp{\bmpsize@temp}{0A}=\z@
```

```
1327            \expandafter\@firstofone
1328          \else
1329            \expandafter\@gobble
1330          \fi
1331          {%
1332            \@bmpsize@break
1333          }%
1334        }%
1335        \edef\bmpsize@pixelheight{\pdf@unescapehex{\bmpsize@tempnum}}%
1336      }{%
1337        \@bmpsize@pushback\bmpsize@head
1338        \@bmpsize@loop{%
1339          \@bmpsize@fillbuf{#1}%
1340          \@bmpsize@grab\bmpsize@temp{1}%
1341          \@bmpsize@skip@one
1342          \edef\bmpsize@off{\the\numexpr\bmpsize@off+1}%
1343          \ifnum\pdf@strcmp{\bmpsize@temp}{0A}=\z@
1344            \expandafter\@firstofone
1345          \else
1346            \expandafter\@gobble
1347          \fi
1348          {%
1349            \@bmpsize@break
1350          }%
1351        }%
1352      }%
1353    }%
1354    }%
1355    }%
1356    }%
1357  }%
1358  \@bmpsize@stop
1359  \@nil
1360  \@bmpsize@end
1361 }%
1362 ⟨/base⟩
```

2.2.8   xpm

```
begin xpm
read 9 0
grab 9 -> $temp
assign {9} -> $offset
check streq $temp ["/* XPM */"]
loop
  fillbuf
  grab 1 -> $temp
  if streq $temp [0x22] % "
    break
  fi
  if streq $temp ["/"]
    fillbuf
    grab 1 -> $temp
    if streq $temp ["*"]
      % look for end of C comment
      loop
        fillbuf
        grab 1 -> $temp
        if streq $temp ["*"]
          loop
            fillbuf
            grab 1 -> $temp
            if streq $temp ["/"]
```

36

```
            break
          fi
          if streq $temp ["*"]
          else
            break
          fi
        repeat
        if streq $temp ["/"]
          break
        fi
      fi
    repeat
  fi
 fi
repeat
% width
assign {} -> $tempnum
loop
  fillbuf
  grab 1 -> $temp
  if iswhite $temp
  else
    if isdigit $temp
      append $tempnum $temp -> $tempnum
      break
    else
      stop
    fi
  fi
repeat
loop
  fillbuf
  grab 1 -> $temp
  if isdigit $temp
    append $tempnum $temp -> $tempnum
  else
    if iswhite $temp
      break
    else
      stop
    fi
  fi
repeat
assign unescapehex($tempnum) -> $pixelwidth
% height
assign {} -> $tempnum
loop
  fillbuf
  grab 1 -> $temp
  if iswhite $temp
  else
    if isdigit $temp
      append $tempnum $temp -> $tempnum
      break
    else
      stop
    fi
  fi
repeat
loop
  fillbuf
  grab 1 -> $temp
```

```
          if isdigit $temp
            append $tempnum $temp -> $tempnum
          else
            if iswhite $temp
              break
            else
              stop
            fi
          fi
        repeat
        assign unescapehex($tempnum) -> $pixelheight
        ok
        end
```

\bmpsize@read@xpm

```
1363 ⟨*base⟩
1364 \def\bmpsize@read@xpm#1{%
1365   \@bmpsize@init
1366   \@bmpsize@read{#1}{9}{0}%
1367   \@bmpsize@grab\bmpsize@temp{9}%
1368   \@bmpsize@skip@four
1369   \@bmpsize@skip@four
1370   \@bmpsize@skip@one
1371   \def\bmpsize@offset{9}%
1372   \ifnum\pdf@strcmp{\bmpsize@temp}{2F2A2058504D202A2F}=\z@
1373   \else
1374     \expandafter\@bmpsize@stop
1375   \fi
1376   \@bmpsize@loop{%
1377     \@bmpsize@fillbuf{#1}%
1378     \@bmpsize@grab\bmpsize@temp{1}%
1379     \@bmpsize@skip@one
1380     \ifnum\pdf@strcmp{\bmpsize@temp}{22}=\z@
1381       \expandafter\@firstofone
1382     \else
1383       \expandafter\@gobble
1384     \fi
1385     {%
1386       \@bmpsize@break
1387     }%
1388     \ifnum\pdf@strcmp{\bmpsize@temp}{2F}=\z@
1389       \expandafter\@firstofone
1390     \else
1391       \expandafter\@gobble
1392     \fi
1393     {%
1394       \@bmpsize@fillbuf{#1}%
1395       \@bmpsize@grab\bmpsize@temp{1}%
1396       \@bmpsize@skip@one
1397       \ifnum\pdf@strcmp{\bmpsize@temp}{2A}=\z@
1398         \expandafter\@firstofone
1399       \else
1400         \expandafter\@gobble
1401       \fi
1402       {%
1403         \@bmpsize@loop{%
1404           \@bmpsize@fillbuf{#1}%
1405           \@bmpsize@grab\bmpsize@temp{1}%
1406           \@bmpsize@skip@one
1407           \ifnum\pdf@strcmp{\bmpsize@temp}{2A}=\z@
1408             \expandafter\@firstofone
1409           \else
1410             \expandafter\@gobble
```

```
1411        \fi
1412        {%
1413          \@bmpsize@loop{%
1414            \@bmpsize@fillbuf{#1}%
1415            \@bmpsize@grab\bmpsize@temp{1}%
1416            \@bmpsize@skip@one
1417            \ifnum\pdf@strcmp{\bmpsize@temp}{2F}=\z@
1418              \expandafter\@firstofone
1419            \else
1420              \expandafter\@gobble
1421            \fi
1422            {%
1423              \@bmpsize@break
1424            }%
1425            \ifnum\pdf@strcmp{\bmpsize@temp}{2A}=\z@
1426              \expandafter\@gobble
1427            \else
1428              \expandafter\@firstofone
1429            \fi
1430            {%
1431              \@bmpsize@break
1432            }%
1433          }%
1434          \ifnum\pdf@strcmp{\bmpsize@temp}{2F}=\z@
1435            \expandafter\@firstofone
1436          \else
1437            \expandafter\@gobble
1438          \fi
1439          {%
1440            \@bmpsize@break
1441          }%
1442        }%
1443      }%
1444    }%
1445  }%
1446 }%
1447 \def\bmpsize@tempnum{}%
1448 \@bmpsize@loop{%
1449   \@bmpsize@fillbuf{#1}%
1450   \@bmpsize@grab\bmpsize@temp{1}%
1451   \@bmpsize@skip@one
1452   \ifcase 0\@bmpsize@iswhite\bmpsize@temp
1453     \expandafter\@gobble
1454   \else
1455     \expandafter\@firstofone
1456   \fi
1457   {%
1458     \ifcase 0\@bmpsize@isdigit\bmpsize@temp
1459       \expandafter\@firstoftwo
1460     \else
1461       \expandafter\@secondoftwo
1462     \fi
1463     {%
1464       \@bmpsize@append\bmpsize@tempnum\bmpsize@tempnum\bmpsize@temp
1465       \@bmpsize@break
1466     }{%
1467       \@bmpsize@stop
1468     }%
1469   }%
1470 }%
1471 \@bmpsize@loop{%
1472   \@bmpsize@fillbuf{#1}%
```

```
1473    \@bmpsize@grab\bmpsize@temp{1}%
1474    \@bmpsize@skip@one
1475    \ifcase 0\@bmpsize@isdigit\bmpsize@temp
1476      \expandafter\@firstoftwo
1477    \else
1478      \expandafter\@secondoftwo
1479    \fi
1480    {%
1481      \@bmpsize@append\bmpsize@tempnum\bmpsize@tempnum\bmpsize@temp
1482    }{%
1483      \ifcase 0\@bmpsize@iswhite\bmpsize@temp
1484        \expandafter\@firstoftwo
1485      \else
1486        \expandafter\@secondoftwo
1487      \fi
1488      {%
1489        \@bmpsize@break
1490      }{%
1491        \@bmpsize@stop
1492      }%
1493    }%
1494 }%
1495 \edef\bmpsize@pixelwidth{\pdf@unescapehex{\bmpsize@tempnum}}%
1496 \def\bmpsize@tempnum{}%
1497 \@bmpsize@loop{%
1498    \@bmpsize@fillbuf{#1}%
1499    \@bmpsize@grab\bmpsize@temp{1}%
1500    \@bmpsize@skip@one
1501    \ifcase 0\@bmpsize@iswhite\bmpsize@temp
1502      \expandafter\@gobble
1503    \else
1504      \expandafter\@firstofone
1505    \fi
1506    {%
1507      \ifcase 0\@bmpsize@isdigit\bmpsize@temp
1508        \expandafter\@firstoftwo
1509      \else
1510        \expandafter\@secondoftwo
1511      \fi
1512      {%
1513        \@bmpsize@append\bmpsize@tempnum\bmpsize@tempnum\bmpsize@temp
1514        \@bmpsize@break
1515      }{%
1516        \@bmpsize@stop
1517      }%
1518    }%
1519 }%
1520 \@bmpsize@loop{%
1521    \@bmpsize@fillbuf{#1}%
1522    \@bmpsize@grab\bmpsize@temp{1}%
1523    \@bmpsize@skip@one
1524    \ifcase 0\@bmpsize@isdigit\bmpsize@temp
1525      \expandafter\@firstoftwo
1526    \else
1527      \expandafter\@secondoftwo
1528    \fi
1529    {%
1530      \@bmpsize@append\bmpsize@tempnum\bmpsize@tempnum\bmpsize@temp
1531    }{%
1532      \ifcase 0\@bmpsize@iswhite\bmpsize@temp
1533        \expandafter\@firstoftwo
1534      \else
```

```
1535        \expandafter\@secondoftwo
1536      \fi
1537      {%
1538        \@bmpsize@break
1539      }{%
1540        \@bmpsize@stop
1541      }%
1542    }%
1543  }%
1544  \edef\bmpsize@pixelheight{\pdf@unescapehex{\bmpsize@tempnum}}%
1545  \@bmpsize@ok
1546  \@bmpsize@stop
1547  \@nil
1548  \@bmpsize@end
1549 }%
1550 ⟨/base⟩
```

2.2.9 tga

```
begin tga
little-endian
                          % id length (1 byte)
read 16 1
grab 1 -> $temp          % color map type (1 byte), values: 0, 1
if streq $temp [0x00]
else
  if streq $temp [0x01]
  else
    stop
  fi
fi
skip 10                  % image type (1 byte)
                         % color map specification (5 bytes)
                         % x origin (2 bytes)
                         % y origin (2 bytes)
num 2 -> $pixelwidth       % image width
num 2 -> $pixelheight      % image height
ok
% TGA File Footer
size 26 -> $temp
read 26 numexpr($temp - 26)
num 4 -> $offset           % the extension area offset
skip 4                     % the developer directory offset
grab 18 -> $temp           % the signature, ".", 0x00
if streq $temp ["TRUEVISION-XFILE." 0x00]
else
  stop
fi
if numeq $offset 0
  stop                     % no extension area
fi
read 4 numexpr($offset + 474) % pixel aspect ratio (4 bytes)
num 2 -> $pixelx           % pixel ratio numerator (pixel width)
num 2 -> $pixely           % pixel ratio denominator (pixel height)
if numeq $pixely 0         % no pixel aspect ratio
  clear $pixelx
  clear $pixely
fi
end
```

\bmpsize@read@tga

```
1551 ⟨*base⟩
1552 \def\bmpsize@read@tga#1{%
```

```
1553  \@bmpsize@init
1554  \@bmpsize@bigendianfalse
1555  \@bmpsize@read{#1}{16}{1}%
1556  \@bmpsize@grab\bmpsize@temp{1}%
1557  \@bmpsize@skip@one
1558  \ifnum\pdf@strcmp{\bmpsize@temp}{00}=\z@
1559    \expandafter\@gobble
1560  \else
1561    \expandafter\@firstofone
1562  \fi
1563  {%
1564    \ifnum\pdf@strcmp{\bmpsize@temp}{01}=\z@
1565      \expandafter\@gobble
1566    \else
1567      \expandafter\@firstofone
1568    \fi
1569    {%
1570      \@bmpsize@stop
1571    }%
1572  }%
1573  \@bmpsize@skip@four
1574  \@bmpsize@skip@four
1575  \@bmpsize@skip@two
1576  \@bmpsize@num@two\bmpsize@pixelwidth
1577  \@bmpsize@num@two\bmpsize@pixelheight
1578  \@bmpsize@ok
1579    \@bmpsize@size{#1}{26}\bmpsize@temp    \@bmpsize@read{#1}{26}{\num-
    expr\bmpsize@temp-26\relax}%
1580  \@bmpsize@num@four\bmpsize@offset
1581  \@bmpsize@skip@four
1582  \@bmpsize@grab\bmpsize@temp{18}%
1583  \@bmpsize@skip@four
1584  \@bmpsize@skip@four
1585  \@bmpsize@skip@four
1586  \@bmpsize@skip@four
1587  \@bmpsize@skip@two
1588  \ifnum\pdf@strcmp{\bmpsize@temp}{54525545564953494F4E2D5846494C452E00}=\z@
1589    \expandafter\@gobble
1590  \else
1591    \expandafter\@firstofone
1592  \fi
1593  {%
1594    \@bmpsize@stop
1595  }%
1596  \ifnum\bmpsize@offset=0\relax
1597    \expandafter\@firstofone
1598  \else
1599    \expandafter\@gobble
1600  \fi
1601  {%
1602    \@bmpsize@stop
1603  }%
1604  \@bmpsize@read{#1}{4}{\numexpr\bmpsize@offset+474\relax}%
1605  \@bmpsize@num@two\bmpsize@pixelx
1606  \@bmpsize@num@two\bmpsize@pixely
1607  \ifnum\bmpsize@pixely=0\relax
1608    \expandafter\@firstofone
1609  \else
1610    \expandafter\@gobble
1611  \fi
1612  {%
1613    \let\bmpsize@pixelx\relax
```

```
1614    \let\bmpsize@pixely\relax
1615    }%
1616    \@bmpsize@stop
1617    \@nil
1618    \@bmpsize@end
1619 }%
1620 ⟨/base⟩
```

2.2.10   pcx

```
begin pcx
little-endian
read 16 0
grab 1 -> $temp          % manufacturer
check streq $temp [0x0A]
skip 1                   % version
num 1 -> $temp           % encoding
check numeq $temp 1
skip 1                   % bits per pixel
num 2 -> $pixelwidth     % x_min
num 2 -> $pixelheight    % y_min
num 2 -> $temp           % x_max
assign numexpr($temp - $pixelwidth + 1) -> $pixelwidth
num 2 -> $temp           % y_max
assign numexpr($temp - $pixelheight + 1) -> $pixelheight
check numgt $pixelwidth 0
check numgt $pixelheight 0
ok
num 2 -> $pixelx         % horizontal resolution in DPI
num 2 -> $pixely         % vertical resolution in DPI
assign {72.27pt} -> $unit
end
```

\bmpsize@read@pcx

```
1621 ⟨*base⟩
1622 \def\bmpsize@read@pcx#1{%
1623   \@bmpsize@init
1624   \@bmpsize@bigendianfalse
1625   \@bmpsize@read{#1}{16}{0}%
1626   \@bmpsize@grab\bmpsize@temp{1}%
1627   \@bmpsize@skip@one
1628   \ifnum\pdf@strcmp{\bmpsize@temp}{0A}=\z@
1629   \else
1630     \expandafter\@bmpsize@stop
1631   \fi
1632   \@bmpsize@skip@one
1633   \@bmpsize@num@one\bmpsize@temp
1634   \ifnum\bmpsize@temp=1\relax
1635   \else
1636     \expandafter\@bmpsize@stop
1637   \fi
1638   \@bmpsize@skip@one
1639   \@bmpsize@num@two\bmpsize@pixelwidth
1640   \@bmpsize@num@two\bmpsize@pixelheight
1641   \@bmpsize@num@two\bmpsize@temp
1642   \edef\bmpsize@pixelwidth{\the\numexpr\bmpsize@temp-\bmpsize@pixelwidth+1}%
1643   \@bmpsize@num@two\bmpsize@temp
1644   \edef\bmpsize@pixelheight{\the\numexpr\bmpsize@temp-\bmpsize@pixelheight+1}%
1645   \ifnum\bmpsize@pixelwidth>0\relax
1646   \else
1647     \expandafter\@bmpsize@stop
1648   \fi
1649   \ifnum\bmpsize@pixelheight>0\relax
```

43

```
1650   \else
1651     \expandafter\@bmpsize@stop
1652   \fi
1653   \@bmpsize@ok
1654   \@bmpsize@num@two\bmpsize@pixelx
1655   \@bmpsize@num@two\bmpsize@pixely
1656   \def\bmpsize@unit{72.27pt}%
1657   \@bmpsize@stop
1658   \@nil
1659   \@bmpsize@end
1660 }%
1661 ⟨/base⟩
```

2.2.11   msp

```
begin msp
little-endian

read 16 0

% header 4
grab 4 -> $temp
if streq $temp ["DanM"]
else
  check streq $temp ["LinS"]
fi
num 2 -> $pixelwidth
num 2 -> $pixelheight
ok
num 2 -> $pixelx % x_asp
num 2 -> $pixely % y_asp
assign {72.27pt} -> $unit % guessing
if numeq $pixelx 0
  num 2 -> $pixelx % x_asp_prn
  num 2 -> $pixely % y_asp_prn
fi
% num 2 % width_prn
% num 2 % height_prn
end
```

\bmpsize@read@msp

```
1662 ⟨*base⟩
1663 \def\bmpsize@read@msp#1{%
1664   \@bmpsize@init
1665   \@bmpsize@bigendianfalse
1666   \@bmpsize@read{#1}{16}{0}%
1667   \@bmpsize@grab\bmpsize@temp{4}%
1668   \@bmpsize@skip@four
1669   \ifnum\pdf@strcmp{\bmpsize@temp}{44616E4D}=\z@
1670     \expandafter\@gobble
1671   \else
1672     \expandafter\@firstofone
1673   \fi
1674   {%
1675     \ifnum\pdf@strcmp{\bmpsize@temp}{4C696E53}=\z@
1676     \else
1677       \expandafter\@bmpsize@stop
1678     \fi
1679   }%
1680   \@bmpsize@num@two\bmpsize@pixelwidth
1681   \@bmpsize@num@two\bmpsize@pixelheight
1682   \@bmpsize@ok
1683   \@bmpsize@num@two\bmpsize@pixelx
```

```
1684    \@bmpsize@num@two\bmpsize@pixely
1685    \def\bmpsize@unit{72.27pt}%
1686    \ifnum\bmpsize@pixelx=0\relax
1687      \expandafter\@firstofone
1688    \else
1689      \expandafter\@gobble
1690    \fi
1691    {%
1692      \@bmpsize@num@two\bmpsize@pixelx
1693      \@bmpsize@num@two\bmpsize@pixely
1694    }%
1695    \@bmpsize@stop
1696    \@nil
1697    \@bmpsize@end
1698 }%
1699 ⟨/base⟩
```

2.2.12 sgi

```
begin sgi
big-endian
read 10 0
grab 2 -> $temp
check streq $temp [0x01 0xDA]    % magic: 474 decimal
grab 1 -> $temp              % storage: 0 or 1
check numge $temp 0
check numle $temp 1
skip 2                      % bpc, dimension
num 2 -> $pixelwidth
num 2 -> $pixelheight
ok
end
```

\bmpsize@read@sgi

```
1700 ⟨*base⟩
1701 \def\bmpsize@read@sgi#1{%
1702   \@bmpsize@init
1703   \@bmpsize@bigendiantrue
1704   \@bmpsize@read{#1}{10}{0}%
1705   \@bmpsize@grab\bmpsize@temp{2}%
1706   \@bmpsize@skip@two
1707   \ifnum\pdf@strcmp{\bmpsize@temp}{01DA}=\z@
1708   \else
1709     \expandafter\@bmpsize@stop
1710   \fi
1711   \@bmpsize@grab\bmpsize@temp{1}%
1712   \@bmpsize@skip@one
1713   \ifnum\bmpsize@temp<0\relax
1714     \expandafter\@bmpsize@stop
1715   \fi
1716   \ifnum\bmpsize@temp>1\relax
1717     \expandafter\@bmpsize@stop
1718   \fi
1719   \@bmpsize@skip@two
1720   \@bmpsize@num@two\bmpsize@pixelwidth
1721   \@bmpsize@num@two\bmpsize@pixelheight
1722   \@bmpsize@ok
1723   \@bmpsize@stop
1724   \@nil
1725   \@bmpsize@end
1726 }%
1727 ⟨/base⟩
```

2.3 Package **bmpsize**

```
1728 ⟨*package⟩
1729 \ProvidesPackage{bmpsize}%
1730   [2016/05/16 v1.7 Extract size/resolution from bitmap files (HO)]%
1731 \RequirePackage{ifpdf}
1732 \ifpdf
1733   \PackageInfo{bmpsize}{Superseded by pdfTeX in PDF mode}%
1734   \expandafter\endinput
1735 \fi
1736 \RequirePackage{pdftexcmds}[2007/11/11]
1737 \begingroup\expandafter\expandafter\expandafter\endgroup
1738 \expandafter\ifx\csname pdf@filedump\endcsname\relax
1739   \PackageError{bmpsize}{%
1740     You need pdfTeX 1.30.0 or newer%
1741   }{Package loading is aborted.}%
1742   \expandafter\endinput
1743 \fi
1744
1745 \RequirePackage{infwarerr}[2007/09/09]
1746 \RequirePackage{graphics}
```

In case of plain TEX options are not executed and `\KV@err` and `\KV@errx` are undefined.

```
1747 \RequirePackage{keyval}\relax
1748 \expandafter\ifx\csname KV@errx\endcsname\relax
1749   \def\KV@errx#1{%
1750     \@PackageError{keyval}{#1}\@ehc
1751   }%
1752 \fi
1753 \expandafter\ifx\csname KV@err\endcsname\relax
1754   \let\KV@err\KV@errx
1755 \fi
1756 \RequirePackage{bmpsize-base}
1757
1758 \InputIfFileExists{bmpsize-\Gin@driver}{}{}
1759
1760 \define@key{Gin}{bmpsizefast}[true]{%
1761 \expandafter\ifx\csname if#1\expandafter\endcsname\csname iftrue\endcsname
1762     \@bmpsize@fasttrue
1763   \else
1764     \@bmpsize@fastfalse
1765   \fi
1766 }
1767 \define@key{Gin}{resolutionunit}{%
1768   \def\bmpsize@unit@default{#1}%
1769 }
1770 \begingroup
1771   \def\x#1{\endgroup
1772     \define@key{Gin}{resolution}{%
1773       \@bmpsize@read@resolution\@bmpsize@user@resolutiontrue##1#1#1\@nil
1774     }%
1775     \define@key{Gin}{defaultresolution}{%
1776       \@bmpsize@read@resolution\@bmpsize@user@resolutionfalse##1#1#1\@nil
1777     }%
1778 }%
1779 \x{ }
1780 \def\@bmpsize@read@resolution#1#2 #3 #4\@nil{%
1781   \ifcase 0\ifx\\#2\\1\fi
1782         \ifnum\pdf@strcmp{#2}{\Gin@exclamation}=\z@
1783           \ifx\\#3\\1\fi
1784           \ifnum\pdf@strcmp{#3}{\Gin@exclamation}=\z@
1785             1%
```

46

```
1786        \fi
1787      \fi
1788    \ifcase\pdf@strcmp{#2}{\Gin@exclamation}\relax
1789      \let\bmpsize@pixelx@default\Gin@exclamation
1790    \else
1791      \edef\bmpsize@pixelx@default{#2}%
1792    \fi
1793    \ifcase\pdf@strcmp{#3}{\Gin@exclamation}\relax
1794      \let\bmpsize@pixely@default\Gin@exclamation
1795    \else
1796      \ifx\\#3\\%
1797        \let\bmpsize@pixely@default\bmpsize@pixelx@default
1798      \else
1799        \edef\bmpsize@pixely@default{#3}%
1800      \fi
1801    \fi
1802    #1%
1803  \else
1804    \PackageError{bmpsize}{%
1805      Wrong syntax for key (default)resolution%
1806    }{%
1807      See package documentation for correct syntax.%
1808    }%
1809  \fi
1810 }
1811 \newcommand*{\bmpsizesetup}{\setkeys{Gin}}
1812
1813 \let\@bmpsize@org@setfile\Gin@setfile
1814 \def\Gin@setfile#1#2#3{%
1815  \ifcase\pdf@strcmp{#1}{bmp}\relax
1816    \expandafter\@firstofone
1817  \else
1818    \expandafter\@gobble
1819  \fi
1820  {%
1821    \bmpsize@okfalse
1822    \edef\bmpsize@ext{\ifx\Gin@ext\relax\Gin@eext\else\Gin@ext\fi}%
1823    \edef\bmpsize@file{\Gin@base\bmpsize@ext}%
1824    \edef\@bmpsize@temp{\bmpsize@ext}%
1825    \@ifundefined{bmpsize@read@\@bmpsize@temp}{%
1826      \@ifundefined{bmpsize@map@\@bmpsize@temp}{}{%
1827        \expandafter\let\expandafter\@bmpsize@temp
1828        \csname bmpsize@map@\@bmpsize@temp\endcsname
1829      }%
1830    }{}%
1831    \@ifundefined{bmpsize@read@\@bmpsize@temp}{%
1832    }{%
1833      \csname bmpsize@read@\@bmpsize@temp\endcsname\bmpsize@file
1834    }%
1835    \ifbmpsize@ok
1836    \else
1837      \@for\@bmpsize@temp:=\bmpsize@types\do{%
1838        \ifbmpsize@ok
1839        \else
1840          \csname bmpsize@read@\@bmpsize@temp\endcsname\bmpsize@file
1841        \fi
1842      }%
1843    \fi
1844    \ifbmpsize@ok
1845      \ifGin@bbox
1846        \@ifundefined{Gin@vllx}{%
1847          \@PackageWarning{bmpsize}{Explicit bounding box is ignored}%
```

```
1848        }{%
1849          \ifx\Gin@viewport@code\relax
1850            \def\Gin@ollx{0}%
1851            \let\Gin@olly\Gin@ollx
1852            \let\Gin@ourx\bmpsize@width
1853            \let\Gin@oury\bmpsize@height
1854            \let\Gin@vllx\Gin@llx
1855            \let\Gin@vlly\Gin@lly
1856            \let\Gin@vurx\Gin@urx
1857            \let\Gin@vury\Gin@ury
1858            \let\Gin@viewport@code\Gin@viewport
1859            \@PackageWarning{bmpsize}{%
1860              Explicit bounding box replaced by\MessageBreak
1861              viewport setting%
1862            }%
1863          \else
1864            \@PackageWarning{bmpsize}{Explicit bounding box is ignored}%
1865          \fi
1866        }%
1867      \fi
1868      \def\Gin@llx{0}%
1869      \def\Gin@lly{0}%
1870      \let\Gin@urx\bmpsize@width
1871      \let\Gin@ury\bmpsize@height
1872      \Gin@bboxtrue
1873    \else
1874      \PackageInfo{bmpsize}{Unknown image type of \bmpsize@file}%
1875    \fi
1876  }%
1877  \@bmpsize@org@setfile{#1}{#2}{#3}%
1878 }
1879 \newcommand*{\bmpsize@ext@type}[1]{%
1880  \@namedef{bmpsize@map@#1}%
1881 }
1882 \bmpsize@ext@type{.jpg}{jpg}
1883 \bmpsize@ext@type{.jpe}{jpg}
1884 \bmpsize@ext@type{.jfif}{jpg}
1885 \bmpsize@ext@type{.jpeg}{jpg}
1886 \bmpsize@ext@type{.tif}{tiff}
1887 \bmpsize@ext@type{.tiff}{tiff}
1888 \bmpsize@ext@type{.pcx}{pcx}
1889 \bmpsize@ext@type{.msp}{msp}
1890 \bmpsize@ext@type{.bmp}{bmp}
1891 \bmpsize@ext@type{.png}{png}
1892 \bmpsize@ext@type{.pnm}{pnm}
1893 \bmpsize@ext@type{.pbm}{pnm}
1894 \bmpsize@ext@type{.pgm}{pnm}
1895 \bmpsize@ext@type{.ppm}{pnm}
1896 \bmpsize@ext@type{.pam}{pam}
1897 \bmpsize@ext@type{.xpm}{xpm}
1898 \bmpsize@ext@type{.gif}{gif}
1899 \bmpsize@ext@type{.tga}{tga}
1900 \bmpsize@ext@type{.sgi}{sgi}
1901 ⟨/package⟩
```

2.4 Drivers

2.4.1 dvips

Identification.

```
1902 ⟨*dvips⟩
1903 \ProvidesFile{bmpsize-dvips.def}%
1904   [2016/05/16 v1.7 Graphics bitmap driver for dvips (HO)]%
```

Ensure correct catcodes.

```
1905 \expandafter\edef\csname @bmpsize@driver@catcodes\endcsname{%
1906   \catcode44 \the\catcode44 % ,
1907   \catcode58 \the\catcode58 % :
1908   \catcode60 \the\catcode60 % <
1909   \catcode61 \the\catcode61 % =
1910   \catcode62 \the\catcode62 % >
1911   \catcode64 \the\catcode64 % @
1912 }
1913 \catcode64 11 %
1914 \@makeother\,
1915 \@makeother\:
1916 \@makeother\<
1917 \@makeother\=
1918 \@makeother\>
```

\Ginclude@bmp Added features: support for viewport/trim and clip.

```
1919 \def\Ginclude@bmp#1{%
1920   \message{<#1>}%
1921   \raise\Gin@req@height
1922   \hbox to\Gin@req@width{%
```

Clipping support.

```
1923     \ifGin@clip
1924       \vbox to\z@{%
1925         \special{ps:gsave currentpoint}%
1926         \kern\Gin@req@height
1927         \hbox to\z@{%
1928           \kern\Gin@req@width
1929           \special{ps:%
1930             currentpoint %
1931             newpath %
1932             3 index 3 index moveto %
1933             1 index 3 index lineto %
1934             2 copy lineto %
1935             exch pop exch pop %
1936             lineto %
1937             closepath %
1938             clip %
1939           }%
1940           \hss
1941         }%
1942         \vss
1943       }%
1944     \fi
```

Support for viewport/trim. The original bounding box is '0 0 width height'. If
package bmpsize is used and the image has been recognized, then the original
width and height are known (\bmpsize@width, \bmpsize@height). Otherwise we
try the saved values \Gin@ourx and \Gin@oury. This guessing will fail, if options
viewport and trim are used both or several times. This is a deficiency of package
graphicx. One of options viewport and trim should be used at most once.

```
1945     \@ifundefined{Gin@ollx}{%
1946       \dimen@\z@
1947     }{%
1948       \ifx\Gin@scalex\Gin@exclamation
1949         \let\Gin@scalex\Gin@scaley
1950       \fi
1951       \ifx\Gin@scaley\Gin@exclamation
1952         \let\Gin@scaley\Gin@scalex
1953       \fi
1954       \@ifundefined{bmpsize@width}{%
1955         \let\bmpsize@width\Gin@ourx
```

```
1956        \let\bmpsize@height\Gin@oury
1957      }{}%
1958      \dimen@=\Gin@llx bp\relax
1959      \dimen@=\Gin@scalex\dimen@
1960      \kern-\dimen@
1961      \advance\Gin@req@width\dimen@
1962      \dimen@=\bmpsize@width bp\relax
1963      \advance\dimen@ by -\Gin@urx bp\relax
1964      \dimen@=\Gin@scalex\dimen@
1965      \advance\Gin@req@width\dimen@
1966      \dimen@=\Gin@lly bp\relax
1967      \dimen@=\Gin@scaley\dimen@
1968      \advance\Gin@req@height\dimen@
1969      \dimen@=\bmpsize@height bp\relax
1970      \advance\dimen@ by -\Gin@ury bp\relax
1971      \dimen@=\Gin@scaley\dimen@
1972      \advance\Gin@req@height\dimen@
1973    }%
1974    \ifdim\dimen@=\z@
1975    \else
1976      \vbox to\z@\bgroup
1977        \kern-\dimen@
1978    \fi
```

The special for the image.

```
1979    \special{em:graph #1,\the\Gin@req@width,\the\Gin@req@height}%
1980    \ifdim\dimen@=\z@
1981    \else
1982      \vss
1983    \egroup
1984    \fi
1985    \ifGin@clip
1986      \special{ps::grestore}%
1987    \fi
1988    \hss
1989  }%
1990 }

1991 \@bmpsize@driver@catcodes
1992 ⟨/dvips⟩
```

2.4.2   dvipdfm and dvipdfmx

Identification.

```
1993 ⟨*dvipdfm⟩
1994 \ProvidesFile{bmpsize-dvipdfm.def}%
1995   [2016/05/16 v1.7 Graphics bitmap driver for dvipdfm (HO)]%
1996 ⟨/dvipdfm⟩
1997 ⟨*dvipdfmx⟩
1998 \ProvidesFile{bmpsize-dvipdfmx.def}%
1999   [2016/05/16 v1.7 Graphics bitmap driver for dvipdfmx (HO)]%
2000 ⟨/dvipdfmx⟩
2001 ⟨*dvipdfm j dvipdfmx⟩
```

Ensure correct catcodes.

```
2002 \expandafter\edef\csname @bmpsize@driver@catcodes\endcsname{%
2003   \catcode44 \the\catcode44 % ,
2004   \catcode46 \the\catcode46 % .
2005   \catcode58 \the\catcode58 % :
2006   \catcode60 \the\catcode60 % <
2007   \catcode61 \the\catcode61 % =
2008   \catcode62 \the\catcode62 % >
2009   \catcode64 \the\catcode64 % @
```

```
2010 }
2011 \catcode64 11 %
2012 \@makeother\,
2013 \@makeother\.
2014 \@makeother\:
2015 \@makeother\<
2016 \@makeother\=
2017 \@makeother\>
```

Counter resource to generate unique names for xform objects.

```
2018 \@ifundefined{@bmpsize@count}{%
2019   \csname newcount\endcsname\@bmpsize@count
2020   \@bmpsize@count=\z@
2021 }{}
```

The file name is given as PDF string in the image special. If we have pdfTEX with \pdfescapestring we use it.

\@bmpsize@pdfescapestring

```
2022 \begingroup\expandafter\expandafter\expandafter\endgroup
2023 \expandafter\ifx\csname pdf@escapestring\endcsname\relax
2024   \def\@bmpsize@pdfescapestring#1{#1}%
2025 \else
2026   \let\@bmpsize@pdfescapestring\pdf@escapestring
2027 \fi
```

The size of reused images of dvipdfm 0.13.2c is 1bp. It is the default size of an image object in user space. Thus the reused image must be scaled to the requested width and height. The factor is just the conversion from pt to bp (72/72.27).

\bmpsize@dvipdfm@factor

```
2028 ⟨dvipdfm⟩\def\bmpsize@dvipdfm@factor{.99626}
```

Unhappily dvipdfmx behaves differently. It remembers the size assuming a resolution of 100 dots per inch and additionally scales the reused image to this size. Thus the scaling factor also depends on the pixel sizes of the image:

- width: $(72 / 72.27) * (100 / 72)$ / pixelwidth =
  $100 / 72.27$ / pixelwidth

- height: $100 / 72.27$ / pixelheight

Recent versions however use the natural size of the reused image. Thus the factor is the difference between the requested size and the natural size.

\Ginclude@bmp Added features: support for viewport/trim, clip, and image reuse.

```
2029 \def\Ginclude@bmp#1{%
2030   \message{<#1>}%
```

Clip support is achieved by putting the image inside a xform object. These xform objects are automatically clipped when they are used.

```
2031 \ifGin@clip
2032   \global\advance\@bmpsize@count\@ne
2033   \edef\@bmpsize@clip@name{@CLIP@\the\@bmpsize@count}%
2034   \special{%
2035     pdf:bxobj \@bmpsize@clip@name\space
2036     width \the\Gin@req@width\space
2037     height \the\Gin@req@height
2038   }%
2039 \fi
```

Support for viewport/trim.

```
2040   \hbox to \z@{%
2041     \@ifundefined{Gin@ollx}{%
2042       \dimen@\z@
```

```
2043  }{%
2044    \ifx\Gin@scalex\Gin@exclamation
2045      \let\Gin@scalex\Gin@scaley
2046    \fi
2047    \ifx\Gin@scaley\Gin@exclamation
2048      \let\Gin@scaley\Gin@scalex
2049    \fi
2050    \@ifundefined{bmpsize@width}{%
2051      \let\bmpsize@width\Gin@ourx
2052      \let\bmpsize@height\Gin@oury
2053    }{}%
2054    \dimen@=\Gin@llx bp\relax
2055    \dimen@=\Gin@scalex\dimen@
2056    \kern-\dimen@
2057    \advance\Gin@req@width\dimen@
2058    \dimen@=\bmpsize@width bp\relax
2059    \advance\dimen@ by -\Gin@urx bp\relax
2060    \dimen@=\Gin@scalex\dimen@
2061    \advance\Gin@req@width\dimen@
2062    \dimen@=\bmpsize@height bp\relax
2063    \advance\dimen@ by -\Gin@ury bp\relax
2064    \dimen@=\Gin@scaley\dimen@
2065    \advance\Gin@req@height\dimen@
2066    \dimen@=\Gin@lly bp\relax
2067    \dimen@=\Gin@scaley\dimen@
2068    \advance\Gin@req@height\dimen@
2069  }%
2070  \ifdim\dimen@=\z@
2071  \else
2072    \vbox to\z@\bgroup
2073      \kern\dimen@
2074  \fi
```

Reuse support, dvipdfm just remember the image. The requested sizes, clipping, …do not matter. In case of dvipdfmx we also must remember the natural size.

```
2075      \edef\@bmpsize@temp{@IMG@\@bmpsize@pdfescapestring{#1}}%
2076      \@ifundefined{\@bmpsize@temp}{%
2077      \global\advance\@bmpsize@count\@ne
2078 ⟨*dvipdfm⟩
2079      \expandafter\xdef\csname\@bmpsize@temp\endcsname{%
2080        \the\@bmpsize@count
2081      }%
2082 ⟨/dvipdfm⟩
2083 ⟨*dvipdfmx⟩
2084      \expandafter\ifx\csname bmpsize@pixelwidth\endcsname\relax
2085      \else
2086        \expandafter\xdef\csname\@bmpsize@temp\endcsname{%
2087          \the\@bmpsize@count:\bmpsize@width:\bmpsize@height
2088        }%
2089      \fi
2090 ⟨/dvipdfmx⟩
2091      \special{%
2092        pdf:image @IMG\the\@bmpsize@count\space
2093        width \the\Gin@req@width\space
2094        height \the\Gin@req@height\space
2095        depth 0pt (\@bmpsize@pdfescapestring{#1})%
2096      }%
2097    }{%
2098 ⟨*dvipdfm⟩
2099      \special{%
2100        pdf:bt %
2101        xscale \strip@pt\dimexpr
2102          \bmpsize@dvipdfm@factor\Gin@req@width\relax\space
```

52

```
2103      yscale \strip@pt\dimexpr
2104        \bmpsize@dvipdfm@factor\Gin@req@height\relax
2105      }%
2106      \special{pdf:uxobj @IMG\csname\@bmpsize@temp\endcsname}%
2107      \special{pdf:et}%
2108 ⟨/dvipdfm⟩
2109 ⟨*dvipdfmx⟩
2110      \expandafter\expandafter\expandafter\@bmpsize@extract
2111        \csname\@bmpsize@temp\endcsname\@nil
2112      \edef\@bmpsize@xscale{\strip@pt\Gin@req@width}%
2113      \edef\@bmpsize@temp{\strip@pt\dimexpr\@bmpsize@width bp}%
2114      \@bmpsize@div\@bmpsize@xscale\@bmpsize@xscale\@bmpsize@temp
2115      \edef\@bmpsize@yscale{\strip@pt\Gin@req@height}%
2116      \edef\@bmpsize@temp{\strip@pt\dimexpr\@bmpsize@height bp}%
2117      \@bmpsize@div\@bmpsize@yscale\@bmpsize@yscale\@bmpsize@temp
2118      \special{%
2119       pdf:bt %
2120       xscale \@bmpsize@xscale\space
2121       yscale \@bmpsize@yscale
2122      }%
2123      \special{pdf:uxobj @IMG\@bmpsize@imgnum}%
2124      \special{pdf:et}%
2125 ⟨/dvipdfmx⟩
2126      }%
2127      \ifdim\dimen@=\z@
2128      \else
2129        \vss
2130      \egroup
2131      \fi
2132      \hss
2133  }%
2134  \ifGin@clip
2135    \special{pdf:exobj}%
2136    \special{pdf:uxobj \@bmpsize@clip@name}%
2137  \fi
2138 }

2139 ⟨*dvipdfmx⟩
2140 \def\@bmpsize@extract#1:#2:#3\@nil{%
2141  \def\@bmpsize@imgnum{#1}%
2142  \def\@bmpsize@width{#2}%
2143  \def\@bmpsize@height{#3}%
2144 }
2145 ⟨/dvipdfmx⟩

2146 \@bmpsize@driver@catcodes
2147 ⟨/dvipdfm j dvipdfmx⟩
```

2.5 Test program **bmpsize-test.tex**

```
2148 ⟨*test⟩
2149 \expandafter\ifx\csname NeedsTeXFormat\endcsname\relax
2150   \input miniltx\relax
2151 \fi
2152 \begingroup\expandafter\expandafter\expandafter\endgroup
2153 \expandafter\ifx\csname pdfoutput\endcsname\relax
2154 \else
2155   \pdfoutput=0 %
2156 \fi
2157 \RequirePackage{bmpsize}
2158
2159 \endlinechar=-1
2160 \catcode`\@=11
```

```
2161 \def\msg#{\immediate\write16}
2162
2163 \def\init{%
2164   \msg{}%
2165   \msg{File name menu}%
2166   \msg{==============}%
2167   \msg{* Option menu: use `opt' as file name}%
2168   \msg{* Quit program: <return>}%
2169   \msg{}%
2170   \message{Image file name = }%
2171   \read-1 to \imagename
2172   \ifx\imagename\@empty
2173     \expandafter\@firstoftwo
2174   \else
2175     \expandafter\@secondoftwo
2176   \fi
2177   {%
2178     \csname @@end\endcsname
2179     \end
2180   }{%
2181     \ifnum\pdf@strcmp{\imagename}{opt}=\z@
2182       \expandafter\optionmenu
2183     \else
2184       \startimg
2185       \expandafter\init
2186     \fi
2187   }%
2188 }
2189 \def\optionmenu{%
2190   \msg{}%
2191   \msg{Option menu}%
2192   \msg{===========}%
2193   \msg{Current setting:}%
2194   \msg{* bmpsizefast = \if@bmpsize@fast true\else false\fi}%
2195   \msg{* \if@bmpsize@user@resolution\else default\fi resolution = %
2196     \bmpsize@pixelx@default
2197     \space
2198     \bmpsize@pixely@default
2199   }%
2200   \msg{* \if@bmpsize@user@resolution default\fi resolution: not set}%
2201   \msg{* resolutionunit = \bmpsize@unit@default}%
2202   \msg{* Quit option menu: <return>}%
2203   \msg{}%
2204   \message{Options = }%
2205   \read-1 to \options
2206   \ifx\options\empty
2207     \expandafter\init
2208   \else
2209     \edef\@bmpsize@temp{%
2210       \noexpand\setkeys{Gin}{\options}%
2211     }%
2212     \@bmpsize@temp
2213     \expandafter\optionmenu
2214   \fi
2215 }
2216
2217 \def\startimg{%
2218   \let\@found\@empty
2219   \msg{}%
2220   \msg{* File [\imagename]}%
2221   \@for\@type:=\bmpsize@types\do{%
2222     \ifx\@found\@empty
```

```
2223     \csname bmpsize@read@\@type\endcsname\imagename
2224     \ifbmpsize@ok
2225       \let\@found\@type
2226       \msg{\space\space Type: \@type}%
2227       \msg{\space\space Pixel width: \bmpsize@pixelwidth\space px}%
2228       \msg{\space\space Pixel height: \bmpsize@pixelheight\space px}%
2229       \ifx\bmpsize@pixelx\relax
2230       \else
2231         \ifx\bmpsize@unit\relax
2232           \let\@unit@spec\@empty
2233           \def\@ratio@name{Ratio }%
2234         \else
2235           \def\@unit@spec{\space dots per \bmpsize@unit}%
2236           \def\@ratio@name{Density }%
2237         \fi
2238         \msg{\space\space \@ratio@name x: \bmpsize@pixelx\@unit@spec}%
2239         \msg{\space\space \@ratio@name y: \bmpsize@pixely\@unit@spec}%
2240       \fi
2241       \msg{\space\space Width: \bmpsize@width\space bp}%
2242       \msg{\space\space Height: \bmpsize@height\space bp}%
2243       \ifx\bmpsize@orientation\relax
2244       \else
2245         \msg{\space\space Orientation: \bmpsize@orientation}%
2246       \fi
2247     \fi
2248   \fi
2249 }%
2250 \ifx\@found\@empty
2251   \edef\@file@date{\pdf@filemoddate{\imagename}}%
2252   \ifx\@file@date\@empty
2253     \msg{\space\space --> File not found <--}%
2254   \else
2255     \msg{\space\space --> Unknown image type <--}%
2256   \fi
2257 \fi
2258 }
2259
2260 \ifx\noinit!\else\expandafter\init\fi
2261 ⟨/test⟩
```

3 Installation

3.1 Download

Package. This package is available on CTAN[1]:

CTAN:macros/latex/contrib/oberdiek/bmpsize.dtx The source file.

CTAN:macros/latex/contrib/oberdiek/bmpsize.pdf Documentation.

Bundle. All the packages of the bundle 'oberdiek' are also available in a TDS compliant ZIP archive. There the packages are already unpacked and the documentation files are generated. The files and directories obey the TDS standard.

CTAN:install/macros/latex/contrib/oberdiek.tds.zip

TDS refers to the standard "A Directory Structure for TeX Files" (**CTAN:tds/tds.pdf**). Directories with `texmf` in their name are usually organized this way.

---

[1] http://ctan.org/pkg/bmpsize

3.2 Bundle installation

Unpacking. Unpack the `oberdiek.tds.zip` in the TDS tree (also known as `texmf` tree) of your choice. Example (linux):

```
unzip oberdiek.tds.zip -d ~/texmf
```

Script installation. Check the directory `TDS:scripts/oberdiek/` for scripts that need further installation steps. Package `attachfile2` comes with the Perl script `pdfatfi.pl` that should be installed in such a way that it can be called as `pdfatfi`. Example (linux):

```
chmod +x scripts/oberdiek/pdfatfi.pl
cp scripts/oberdiek/pdfatfi.pl /usr/local/bin/
```

3.3 Package installation

Unpacking. The `.dtx` file is a self-extracting `docstrip` archive. The files are extracted by running the `.dtx` through plain TeX:

```
tex bmpsize.dtx
```

TDS. Now the different files must be moved into the different directories in your installation TDS tree (also known as `texmf` tree):

`bmpsize.sty`	→ `tex/latex/oberdiek/bmpsize.sty`
`bmpsize-base.sty`	→ `tex/latex/oberdiek/bmpsize-base.sty`
`bmpsize-test.tex`	→ `tex/latex/oberdiek/bmpsize-test.tex`
`bmpsize-dvips.def`	→ `tex/latex/oberdiek/bmpsize-dvips.def`
`bmpsize-dvipdfm.def`	→ `tex/latex/oberdiek/bmpsize-dvipdfm.def`
`bmpsize-dvipdfmx.def`	→ `tex/latex/oberdiek/bmpsize-dvipdfmx.def`
`bmpsize.pdf`	→ `doc/latex/oberdiek/bmpsize.pdf`
`bmpsize.dtx`	→ `source/latex/oberdiek/bmpsize.dtx`

If you have a `docstrip.cfg` that configures and enables `docstrip`'s TDS installing feature, then some files can already be in the right place, see the documentation of `docstrip`.

3.4 Refresh file name databases

If your TeX distribution (teTeX, mikTeX, …) relies on file name databases, you must refresh these. For example, teTeX users run `texhash` or `mktexlsr`.

3.5 Some details for the interested

Unpacking with LaTeX. The `.dtx` chooses its action depending on the format:

plain TeX: Run `docstrip` and extract the files.

LaTeX: Generate the documentation.

If you insist on using LaTeX for `docstrip` (really, `docstrip` does not need LaTeX), then inform the autodetect routine about your intention:

```
latex \let\install=y\input{bmpsize.dtx}
```

Do not forget to quote the argument according to the demands of your shell.

Generating the documentation. You can use both the `.dtx` or the `.drv` to generate the documentation. The process can be configured by the configuration file `ltxdoc.cfg`. For instance, put this line into this file, if you want to have A4 as paper format:

```
\PassOptionsToClass{a4paper}{article}
```

An example follows how to generate the documentation with pdfLaTeX:

```
pdflatex bmpsize.dtx
makeindex -s gind.ist bmpsize.idx
pdflatex bmpsize.dtx
makeindex -s gind.ist bmpsize.idx
pdflatex bmpsize.dtx
```

4 Catalogue

The following XML file can be used as source for the TeX Catalogue. The elements `caption` and `description` are imported from the original XML file from the Catalogue. The name of the XML file in the Catalogue is `bmpsize.xml`.

```
2262 ⟨*catalogue⟩
2263 <?xml version='1.0' encoding='us-ascii'?>
2264 <!DOCTYPE entry SYSTEM 'catalogue.dtd'>
2265 <entry datestamp='$Date$' modifier='$Author$' id='bmpsize'>
2266   <name>bmpsize</name>
2267   <caption>Extract size and resolution data from bitmap files.</caption>
2268   <authorref id='auth:oberdiek'/>
2269   <copyright owner='Heiko Oberdiek' year='2006-2009'/>
2270   <license type='lppl1.3'/>
2271   <version number='1.7'/>
2272   <description>
2273     This package analyzes bitmap images to extract size and resolution
2274     data.  It adds this feature to the graphics package so it is no
2275     longer necessary to provide a separate bounding box files for
2276     bitmap images.  dditionally the implementation for the inclusion
2277     of bitmap images in some drivers of package
2278     <xref refid='graphicx'>graphicx</xref> are rewritten to support
2279     options viewport, trim and clip.  The package requires
2280     <xref refid='pdftex'>pdfTeX</xref> version 1.30.0 or later (the
2281     relevant pdfTeX primitive operates in both DVI and PDF output
2282     modes).
2283     <p/>
2284     The package is part of the <xref refid='oberdiek'>oberdiek</xref>
2285     bundle.
2286   </description>
2287   <documentation details='Package documentation'
2288       href='ctan:/macros/latex/contrib/oberdiek/bmpsize.pdf'/>
2289   <ctan file='true' path='/macros/latex/contrib/oberdiek/bmpsize.dtx'/>
2290   <miktex location='oberdiek'/>
2291   <texlive location='oberdiek'/>
2292   <install path='/macros/latex/contrib/oberdiek/oberdiek.tds.zip'/>
2293 </entry>
2294 ⟨/catalogue⟩
```

5 References

[1] D. P. Carlisle, The LaTeX Project: *Packages in the 'graphics' bundle*, 2005/11/14; CTAN:macros/latex/required/graphics/grfguide.pdf.

57

5.1 URLs for bitmap format descriptions

5.1.1 JPEG

- http://www.w3.org/Graphics/JPEG/jfif3.pdf
- http://exif.org/Exif2-2.PDF

5.1.2 PNG

- http://en.wikipedia.org/wiki/PNG
- http://www.w3.org/TR/PNG/

5.1.3 GIF

- http://www.w3.org/Graphics/GIF/spec-gif89a.txt

5.1.4 BMP

- http://en.wikipedia.org/wiki/Windows_bitmap
- http://de.wikipedia.org/wiki/Windows_bitmap
- http://msdn.microsoft.com/en-us/library/ms532311.aspx
- http://msdn.microsoft.com/en-us/library/ms532321.aspx

5.1.5 PCX

- http://en.wikipedia.org/wiki/PCX
- http://de.wikipedia.org/wiki/PCX
- http://www.qzx.com/pc-gpe/pcx.txt

5.1.6 MSP

- http://en.wikipedia.org/wiki/Microsoft_Paint
- Sources of dvips.

5.1.7 TIFF

- http://en.wikipedia.org/wiki/TIFF
- http://partners.adobe.com/public/developer/en/tiff/TIFF6.pdf

5.1.8 TGA

- http://de.wikipedia.org/wiki/Targa_Image_File
- http://en.wikipedia.org/wiki/Truevision_TGA
- http://www.dca.fee.unicamp.br/~martino/disciplinas/ea978/tgaffs.pdf

5.1.9 SGI

- http://en.wikipedia.org/wiki/Silicon_Graphics_Image
- ftp://ftp.sgi.com/graphics/SGIIMAGESPEC

5.1.10 WMF

- <http://www.fileformat.info/format/wmf/>

5.1.11 XPM

- <http://en.wikipedia.org/wiki/XPM_%28image_format%29>
- <http://de.wikipedia.org/wiki/Xpm>
- <http://koala.ilog.fr/ftp/pub/xpm/xpm-README.html>

6 History

[2006/08/24 v1.0]

- First version.

[2007/02/18 v1.1]

- 1in replaced by 72.27pt, because TeX is inaccurate if 1in is given.

[2007/04/11 v1.2]

- Line ends sanitized.

[2007/05/01 v1.3]

- Uses package infwarerr.
- Image reuse algorithm fixed for dvipdfmx.
- Some support for Exif's orientation tag.

[2007/11/11 v1.4]

- Use of package pdftexcmds for LuaTeX support.
- Fix of bug of package keyval: \KV@err and \KV@errx are used, but undefined if loaded by plain TeX.

[2008/08/11 v1.5]

- Code is not changed.
- Update of URLs.

[2009/09/04 v1.6]

- Fixes for reusing objects with dvipdfmx-20090708. Older versions of dvipdfmx are no longer supported.

[2016/05/16 v1.7]

- Documentation updates.

7 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; plain numbers refer to the code lines where the entry is used.

60

61

62

878, 1003, 1009, 1012, 1035,
1045, 1053, 1150, 1166, 1178,
1193, 1248, 1262, 1271, 1326,
1343, 1372, 1380, 1388, 1397,
1407, 1417, 1425, 1434, 1558,

1564, 1588, 1628, 1669, 1675,
1707, 1782, 1784, 1924, 1927,
1946, 1974, 1976, 1980, 2020,
2040, 2042, 2070, 2072, 2127, 2181