The **bmpsize** package

Heiko Oberdiek

<oberdiek@uni-freiburg.de>

2006/08/24 v1.0

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

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 1in. Thus the correct driver option must be given. dvipdfm and dvipdfmx are not equivalent.

2 Implementation

2.1 Basic package **bmpsize-base**

Identification.

```
1 ⟨*base⟩
2 \ProvidesPackage{bmpsize-base}%
3    [2006/08/24 v1.0 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.

\@bmpsize@plain@loop

```
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 \newif\ifbmpsize@ok
15 \let\@bmpsize@ok\bmpsize@oktrue
16
17 \newif\if@bmpsize@bigendian
18 \newif\if@bmpsize@absnum
19 \newif\if@bmpsize@user@resolution
20 \newif\if@bmpsize@fast
21 \@bmpsize@fasttrue
```

```
22
23 \def\@bmpsize@init{%
24   \let\@bmpsize@org@plain@loop\loop
25   \let\loop\@bmpsize@plain@loop
26   \bmpsize@okfalse
27   \@bmpsize@bigendiantrue
28   \@bmpsize@absnumfalse
29   \let\bmpsize@pixelwidth\relax
30   \let\bmpsize@pixelheight\relax
31   \let\bmpsize@pixelx\relax
32   \let\bmpsize@pixely\relax
33   \let\bmpsize@unit\relax
34   \let\bmpsize@pixelxdenom\relax
35   \let\bmpsize@pixelydenom\relax
36 }
37
38 \def\@bmpsize@stop#1\@nil{}
39
40 \def\@bmpsize@loop#1{%
41   #1%
42   \@bmpsize@loop{#1}%
43 }
44 \def\@bmpsize@break#1\@bmpsize@loop#2{}
45
46 \def\@bmpsize@size#1#2#3{%
47   \edef#3{\pdffilesize{#1}}%
48   \ifx#3\@empty
49     \expandafter\@bmpsize@stop
50   \fi
51   \ifnum#3<#2\relax
52     \expandafter\@bmpsize@stop
53   \fi
54 }
55
56 \def\@bmpsize@read#1#2#3{%
57   \edef\@bmpsize@buf{\pdffiledump offset#3length#2{#1}}%
58   \edef\@bmpsize@temp{%
59     \noexpand\@bmpsize@check@byte{#2}\@bmpsize@buf{}{}\noexpand\\%
60   }%
61   \@bmpsize@temp
62 }
63 \def\@bmpsize@fillbuf#1{%
64   \ifx\@bmpsize@buf\@empty
65     \expandafter\@firstofone
66   \else
67     \expandafter\@gobble
68   \fi
69   {%
70     \edef\@bmpsize@buf{%
71       \pdffiledump offset\bmpsize@offset
72                    length\bmpsize@fillbuflength{#1}%
73     }%
74     \ifx\@bmpsize@buf\@empty
75       \expandafter\@bmpsize@stop
76     \fi
77     \edef\bmpsize@offset{\the\numexpr\bmpsize@offset+\bmpsize@fillbuflength}%
78   }%
79 }
80 \def\bmpsize@fillbuflength{10}
81
82 \def\@bmpsize@append#1#2#3{%
83   \edef#1{#2#3}%
```

```latex
84 }
85 \def\@bmpsize@pushback#1{%
86   \edef\@bmpsize@buf{#1\@bmpsize@buf}%
87 }
88
89 \def\@bmpsize@iswhite#1{%
90   \ifnum\pdfstrcmp{#1}{09}=\z@
91   \else
92     \ifnum\pdfstrcmp{#1}{0A}=\z@
93     \else
94       \ifnum\pdfstrcmp{#1}{0D}=\z@
95       \else
96         \ifnum\pdfstrcmp{#1}{20}=\z@
97         \else
98           1%
99         \fi
100       \fi
101     \fi
102   \fi
103   \space
104 }
105 \def\@bmpsize@isdigit#1{%
106   \ifnum\pdfstrcmp{#1}{30}<\z@
107     1%
108   \else
109     \ifnum\pdfstrcmp{#1}{39}>\z@
110       1%
111     \fi
112   \fi
113   \space
114 }
115
116 \def\@bmpsize@check@byte#1#2#3{%
117   \ifnum#1<\@ne
118     \csname fi\endcsname
119     \@bmpsize@cleanup@end
120   \else
121     \csname fi\endcsname
122   \ifx!#2#3!%
123     \csname fi\endcsname
124     \@bmpsize@stop
125   \else
126     \csname fi\endcsname
127     \expandafter\@bmpsize@check@byte\expandafter{\the\numexpr#1-1}%
128 }
129 \def\@bmpsize@cleanup@end#1\\{}
130
131 \def\@bmpsize@swap@maybe#1{%
132   \if@bmpsize@bigendian
133   \else
134     \edef#1{\expandafter\@bmpsize@@swap#1\@empty\@empty\@empty\@empty}%
135   \fi
136 }
137 \def\@bmpsize@@swap#1#2#3#4#5#6#7#8{%
138   #7#8#5#6#3#4#1#2%
139 }
140
141 \def\@bmpsize@skip@one{%
142   \edef\@bmpsize@buf{\expandafter\@gobbletwo\@bmpsize@buf}%
143 }
144 \def\@bmpsize@skip@two{%
145   \edef\@bmpsize@buf{\expandafter\@gobblefour\@bmpsize@buf}%
```

```
146 }
147 \def\@bmpsize@skip@four{%
148   \edef\@bmpsize@buf{%
149     \expandafter\expandafter\expandafter\@gobblefour\expandafter
150     \@gobblefour\@bmpsize@buf
151   }%
152 }
153
154 \def\@bmpsize@grab#1#2{%
155   \edef#1{\noexpand\@bmpsize@grab@byte#2=\@bmpsize@buf\noexpand\\}%
156   \edef#1{#1}%
157 }
158 \def\@bmpsize@grab@byte#1=#2#3{%
159   #2#3%
160   \ifnum#1>\@ne
161     \expandafter\@bmpsize@grab@byte\the\numexpr#1-1\expandafter=%
162   \else
163     \expandafter\@bmpsize@cleanup@end
164   \fi
165 }
166
167 \def\@bmpsize@abs@maybe#1{%
168   \let\@bmpsize@temp\relax
169   \if@bmpsize@absnum
170     \ifnum"\expandafter\@car#1\@nil>7 %
171       \edef#1{\expandafter\@bmpsize@abs@byte#1\relax}%
172       \ifnum\pdfstrcmp{#1}{7FFFFFFF}=\z@
173         \let\@bmpsize@temp\@bmpsize@stop
174       \else
175         \def\@bmpsize@temp{\edef#1{\the\numexpr#1+1}}%
176       \fi
177     \fi
178   \fi
179 }
180 \def\@bmpsize@abs@byte#1{%
181   \ifx#1\relax
182   \else
183     \ifcase"0#1 %
184       F\or E\or D\or C\or B\or A\or 9\or 8\or
185       7\or 6\or 5\or 4\or 3\or 2\or 1\or 0%
186     \fi
187     \expandafter\@bmpsize@abs@byte
188   \fi
189 }
190
191 \def\@bmpsize@num@one#1{%
192   \@bmpsize@grab#11%
193   \@bmpsize@abs@maybe#1%
194   \edef#1{\number"#1}%
195   \@bmpsize@temp
196   \@bmpsize@skip@one
197 }
198 \def\@bmpsize@num@two#1{%
199   \@bmpsize@grab#12%
200   \@bmpsize@swap@maybe#1%
201   \@bmpsize@abs@maybe#1%
202   \edef#1{\number"#1}%
203   \@bmpsize@temp
204   \@bmpsize@skip@two
205 }
206 \def\@bmpsize@num@four#1{%
207   \@bmpsize@grab#14%
```

```
208    \@bmpsize@swap@maybe#1%
209    \@bmpsize@abs@maybe#1%
210    \ifnum\pdfstrcmp{#1}{7FFFFFFF}>\z@
211      \expandafter\@bmpsize@stop
212    \fi
213    \edef#1{\number"#1}%
214    \@bmpsize@temp
215    \@bmpsize@skip@four
216 }
217
218 \def\@bmpsize@div#1#2#3{% #1 := #2/#3
219    \FPdiv#1{#2}{#3}%
220    \@bmpsize@beautify#1%
221 }
222 \def\@bmpsize@beautify#1{%
223    \FPifint#1%
224      \edef#1{\expandafter\@bmpsize@trunc#1.\@nil}%
225    \else
226      \edef#1{\expandafter\@bmpsize@cleanup@frac#1.\@nil}%
227    \fi
228 }
229 \def\@bmpsize@trunc#1.#2\@nil{#1}
230 % #1 isn't an integer, thus we should have at least one
231 % necessary digit after the dot
232 \def\@bmpsize@cleanup@frac#1.#2#3.#4\@nil{%
233    #1.#2%
234    \ifx\\#3\\%
235    \else
236      \@bmpsize@cleanup@fracdigits#3000000000\@nil
237    \fi
238 }
239 \def\@bmpsize@cleanup@fracdigits#1#2#3#4#5#6#7#8#9{%
240    \ifcase#9 %
241      \ifcase#8 %
242        \ifcase#7 %
243          \ifcase#6 %
244            \ifcase#5 %
245              \ifcase #4 %
246                \ifcase #3 %
247                  \ifcase #2 %
248                    \ifcase #1 %
249                    \else
250                      #1%
251                    \fi
252                  \else
253                    #1#2%
254                  \fi
255                \else
256                  #1#2#3%
257                \fi
258              \else
259                #1#2#3#4%
260              \fi
261            \else
262              #1#2#3#4#5%
263            \fi
264          \else
265            #1#2#3#4#5#6%
266          \fi
267        \else
268          #1#2#3#4#5#6#7%
269        \fi
```

```
270        \else
271          #1#2#3#4#5#6#7#8%
272        \fi
273      \else
274        #1#2#3#4#5#6#7#8#9%
275      \fi
276      \@bmpsize@trunc.%
277 }
278
279 \def\@bmpsize@end{%
280    \ifbmpsize@ok
281      \ifx\bmpsize@pixelwidth\relax
282        \bmpsize@okfalse
283      \fi
284      \ifx\bmpsize@pixelheight\relax
285        \bmpsize@okfalse
286      \fi
287    \fi
288    \ifbmpsize@ok
289      \ifnum\bmpsize@pixelwidth>\z@
290      \else
291        \bmpsize@okfalse
292      \fi
293      \ifnum\bmpsize@pixelheight>\z@
294      \else
295        \bmpsize@okfalse
296      \fi
297    \fi
298    \ifbmpsize@ok
299      \ifcase 0%
300        \ifx\bmpsize@pixelx\relax 1 \fi
301        \ifx\bmpsize@pixely\relax 1 \fi
302        \ifnum\bmpsize@pixelx>\z@\else 1 \fi
303        \ifnum\bmpsize@pixely>\z@\else 1 \fi
304        \ifx\bmpsize@pixelxdenom\relax
305          \ifx\bmpsize@pixelydenom\relax\else 1 \fi
306        \else
307          \ifnum\bmpsize@pixelxdenom>\z@\else 1 \fi
308        \fi
309        \ifx\bmpsize@pixelydenom\relax
310        \else
311          \ifnum\bmpsize@pixelydenom>\z@\else 1 \fi
312        \fi
313      \else
314        \let\bmpsize@pixelx\relax
315        \let\bmpsize@pixely\relax
316        \let\bmpsize@unit\relax
317        \let\bmpsize@pixelxdenom\relax
318        \let\bmpsize@pixelydenom\relax
319      \fi
320      \ifx\bmpsize@pixelxdenom\relax
321      \else
322        \@bmpsize@div\bmpsize@pixelx\bmpsize@pixelx\bmpsize@pixelxdenom
323        \@bmpsize@div\bmpsize@pixely\bmpsize@pixely\bmpsize@pixelydenom
324        \let\bmpsize@pixelxdenom\relax
325        \let\bmpsize@pixelydenom\relax
326      \fi
327      \ifcase 0\ifx\bmpsize@unit\relax 1\fi
328              \if@bmpsize@user@resolution 1\fi
329              \relax
330        \let\bmpsize@calc@unit\bmpsize@unit
331        \let\bmpsize@calc@pixelx\bmpsize@pixelx
```

```
332        \let\bmpsize@calc@pixely\bmpsize@pixely
333      \else
334        \let\bmpsize@calc@unit\bmpsize@unit@default
335        \let\bmpsize@calc@pixelx\bmpsize@pixelx@default
336        \let\bmpsize@calc@pixely\bmpsize@pixely@default
337        \ifx\bmpsize@calc@pixely\Gin@exclamation
338          \ifx\bmpsize@pixelx\relax
339            \let\bmpsize@calc@pixely\bmpsize@calc@pixelx
340          \else
341            \FPdiv\bmpsize@calc@pixely\bmpsize@calc@pixelx\bmpsize@pixelx
342            \FPmul\bmpsize@calc@pixely\bmpsize@calc@pixely\bmpsize@pixely
343          \fi
344        \else
345          \ifx\bmpsize@calc@pixelx\Gin@exclamation
346            \ifx\bmpsize@pixelx\relax
347              \let\bmpsize@calc@pixelx\bmpsize@calc@pixely
348            \else
349              \FPdiv\bmpsize@calc@pixelx\bmpsize@calc@pixely\bmpsize@pixely
350              \FPmul\bmpsize@calc@pixelx\bmpsize@calc@pixelx\bmpsize@pixelx
351            \fi
352          \fi
353        \fi
354      \fi
355      \FPdiv\bmpsize@width\bmpsize@pixelwidth\bmpsize@calc@pixelx
356      \FPdiv\bmpsize@height\bmpsize@pixelheight\bmpsize@calc@pixely
357      % calculation of width and height in bp for package graphics
358      % 1in = 72bp = 72.27pt, 72/72.27 = 8/8.03, 1pt = 65536sp
359      \if@bmpsize@fast
360        \edef\bmpsize@width{%
361          \strip@pt\dimexpr.99626\dimexpr
362          \bmpsize@width\dimexpr\bmpsize@calc@unit
363        }%
364        \edef\bmpsize@height{%
365          \strip@pt\dimexpr.99626\dimexpr
366          \bmpsize@height\dimexpr\bmpsize@calc@unit
367        }%
368      \else
369        \edef\@bmpsize@temp{\number\dimexpr\bmpsize@calc@unit}%
370        \ifnum\@bmpsize@temp>100000 %
371          \FPmul\@bmpsize@temp\@bmpsize@temp{0.00001}%
372          \def\@bmpsize@corr{100000}%
373        \else
374          \let\@bmpsize@corr\relax
375        \fi
376        \FPmul\bmpsize@width\bmpsize@width\@bmpsize@temp
377        \FPmul\bmpsize@height\bmpsize@height\@bmpsize@temp
378        \FPmul\bmpsize@width\bmpsize@width{8}%
379        \FPmul\bmpsize@height\bmpsize@height{8}%
380        \FPdiv\bmpsize@width\bmpsize@width{8.03}%
381        \FPdiv\bmpsize@height\bmpsize@height{8.03}%
382        \FPdiv\bmpsize@width\bmpsize@width{65536}%
383        \FPdiv\bmpsize@height\bmpsize@height{65536}%
384        \ifx\@bmpsize@corr\relax
385        \else
386          \FPmul\bmpsize@width\bmpsize@width\@bmpsize@corr
387          \FPmul\bmpsize@height\bmpsize@height\@bmpsize@corr
388        \fi
389        \FPround\bmpsize@width\bmpsize@width{5}%
390        \FPround\bmpsize@height\bmpsize@height{5}%
391        \@bmpsize@beautify\bmpsize@width
392        \@bmpsize@beautify\bmpsize@height
393      \fi
```

```
394   \fi
395   \let\loop\@bmpsize@org@plain@loop
396 }
397 \def\bmpsize@unit@default{1in}
398 \def\bmpsize@pixelx@default{72}
399 \let\bmpsize@pixely@default\Gin@exclamation
400
401 \def\bmpsize@types{png,jpg,bmp,gif,tiff,pnm,pam,xpm,tga,pcx,msp,sgi}
402 ⟨/base⟩
```

2.2   Bitmap formats

2.2.1   png

¡*ignore¿

```
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
```

¡/ignore¿

\bmpsize@read@png

```
403 ⟨*base⟩
404 \def\bmpsize@read@png#1{%
405   \@bmpsize@init
406   \@bmpsize@bigendiantrue
407   \@bmpsize@read{#1}{24}{0}%
408   \@bmpsize@grab\bmpsize@temp{8}%
409   \@bmpsize@skip@four
410   \@bmpsize@skip@four
411   \ifnum\pdfstrcmp{\bmpsize@temp}{89504E470D0A1A0A}=\z@
412   \else
```

```
413        \expandafter\@bmpsize@stop
414    \fi
415    \@bmpsize@num@four\bmpsize@length
416    \@bmpsize@grab\bmpsize@temp{4}%
417    \@bmpsize@skip@four
418    \ifnum\pdfstrcmp{\bmpsize@temp}{49484452}=\z@
419    \else
420        \expandafter\@bmpsize@stop
421    \fi
422    \@bmpsize@num@four\bmpsize@pixelwidth
423    \@bmpsize@num@four\bmpsize@pixelheight
424    \@bmpsize@ok
425    \edef\bmpsize@offset{\the\numexpr20+\bmpsize@length}%
426    \@bmpsize@loop{%
427      \@bmpsize@read{#1}{8}{\bmpsize@offset}%
428      \@bmpsize@num@four\bmpsize@length
429      \@bmpsize@grab\bmpsize@temp{4}%
430      \@bmpsize@skip@four
431      \ifnum\pdfstrcmp{\bmpsize@temp}{49444154}=\z@
432        \expandafter\@firstofone
433      \else
434        \expandafter\@gobble
435      \fi
436      {%
437        \@bmpsize@stop
438      }%
439      \ifnum\pdfstrcmp{\bmpsize@temp}{70485973}=\z@
440        \expandafter\@firstofone
441      \else
442        \expandafter\@gobble
443      \fi
444      {%
445        \@bmpsize@read{#1}{9}{\numexpr\bmpsize@offset+8\relax}%
446        \@bmpsize@num@four\bmpsize@pixelx
447        \@bmpsize@num@four\bmpsize@pixely
448        \@bmpsize@grab\bmpsize@temp{1}%
449        \@bmpsize@skip@one
450        \ifnum\bmpsize@temp=1\relax
451          \expandafter\@firstofone
452        \else
453          \expandafter\@gobble
454        \fi
455        {%
456          \def\bmpsize@unit{100cm}%
457        }%
458        \@bmpsize@stop
459      }%
460      \edef\bmpsize@offset{\the\numexpr\bmpsize@offset+12+\bmpsize@length}%
461    }%
462    \@bmpsize@stop
463    \@nil
464    \@bmpsize@end
465 }%
466 ⟨/base⟩
```

2.2.2 jpg

¡*ignore¿

```
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 {1in} -> $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
          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 {1in} -> $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 282 % XResolution
      skip 6
      num 4 -> $temp
      read 8 numexpr($temp + $exifoffset)
      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 -> $temp
      read 8 numexpr($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
```

¡/ignore¿

\bmpsize@read@jpg

```
467 ⟨∗base⟩
468 \def\bmpsize@read@jpg#1{%
469   \@bmpsize@init
470   \@bmpsize@read{#1}{3}{0}%
471   \@bmpsize@grab\bmpsize@temp{3}%
472   \@bmpsize@skip@two
473   \@bmpsize@skip@one
474   \ifnum\pdfstrcmp{\bmpsize@temp}{FFD8FF}=\z@
475   \else
476     \expandafter\@bmpsize@stop
477   \fi
478   \def\bmpsize@offset{2}%
479   \def\bmpsize@exifdensity{0}%
480   \@bmpsize@loop{%
481     \@bmpsize@read{#1}{4}{\bmpsize@offset}%
```

```
482    \@bmpsize@grab\bmpsize@temp{1}%
483    \@bmpsize@skip@one
484    \ifnum\pdfstrcmp{\bmpsize@temp}{FF}=\z@
485    \else
486      \expandafter\@bmpsize@stop
487    \fi
488    \@bmpsize@num@one\bmpsize@temp
489    \ifnum\bmpsize@temp=218\relax
490      \expandafter\@firstofone
491    \else
492      \expandafter\@gobble
493    \fi
494    {%
495      \@bmpsize@stop
496    }%
497    \ifnum\bmpsize@temp=224\relax
498      \expandafter\@firstoftwo
499    \else
500      \expandafter\@secondoftwo
501    \fi
502    {%
503      \@bmpsize@num@two\bmpsize@length
504      \ifnum\bmpsize@exifdensity=0\relax
505        \expandafter\@firstofone
506      \else
507        \expandafter\@gobble
508      \fi
509      {%
510        \unless\ifnum\bmpsize@length<16\relax
511          \expandafter\@firstofone
512        \else
513          \expandafter\@gobble
514        \fi
515        {%
516          \@bmpsize@read{#1}{12}{\numexpr\bmpsize@offset+4\relax}%
517          \@bmpsize@grab\bmpsize@temp{5}%
518          \@bmpsize@skip@four
519          \@bmpsize@skip@one
520          \ifnum\pdfstrcmp{\bmpsize@temp}{4A46494600}=\z@
521            \expandafter\@firstofone
522          \else
523            \expandafter\@gobble
524          \fi
525          {%
526            \ifnum\bmpsize@length<16\relax
527              \expandafter\@bmpsize@stop
528            \fi
529            \@bmpsize@skip@two
530            \@bmpsize@num@one\bmpsize@temp
531            \ifnum\bmpsize@temp=1\relax
532              \expandafter\@firstoftwo
533            \else
534              \expandafter\@secondoftwo
535            \fi
536            {%
537              \def\bmpsize@unit{1in}%
538            }{%
539              \ifnum\bmpsize@temp=2\relax
540                \expandafter\@firstofone
541              \else
542                \expandafter\@gobble
543              \fi
```

16

```
544              {%
545                \def\bmpsize@unit{1cm}%
546              }%
547            }%
548            \@bmpsize@num@two\bmpsize@pixelx
549            \@bmpsize@num@two\bmpsize@pixely
550          }%
551        }%
552      }%
553    }{%
554      \ifnum\bmpsize@temp=225\relax
555        \expandafter\@firstoftwo
556      \else
557        \expandafter\@secondoftwo
558      \fi
559      {%
560        \@bmpsize@num@two\bmpsize@length
561        \unless\ifnum\bmpsize@length<20\relax
562          \expandafter\@firstofone
563        \else
564          \expandafter\@gobble
565        \fi
566        {%
567          \@bmpsize@read{#1}{20}{\numexpr\bmpsize@offset+4\relax}%
568          \@bmpsize@grab\bmpsize@temp{6}%
569          \@bmpsize@skip@four
570          \@bmpsize@skip@two
571          \ifnum\pdfstrcmp{\bmpsize@temp}{457869660000}=\z@
572            \expandafter\@firstofone
573          \else
574            \expandafter\@gobble
575          \fi
576          {%
577            \edef\bmpsize@exifoffset{\the\numexpr\bmpsize@offset+10}%
578            \@bmpsize@grab\bmpsize@temp{2}%
579            \@bmpsize@skip@two
580            \ifnum\pdfstrcmp{\bmpsize@temp}{4949}=\z@
581              \expandafter\@firstoftwo
582            \else
583              \expandafter\@secondoftwo
584            \fi
585            {%
586              \@bmpsize@bigendianfalse
587            }{%
588              \ifnum\pdfstrcmp{\bmpsize@temp}{4D4D}=\z@
589              \else
590                \expandafter\@bmpsize@stop
591              \fi
592            }%
593            \@bmpsize@num@two\bmpsize@temp
594            \ifnum\bmpsize@temp=42\relax
595            \else
596              \expandafter\@bmpsize@stop
597            \fi
598            \@bmpsize@num@four\bmpsize@temp
599            \ifnum\bmpsize@temp>0\relax
600            \else
601              \expandafter\@bmpsize@stop
602            \fi
603            \edef\bmpsize@off{\the\numexpr\bmpsize@temp+\bmpsize@exifoffset}%
604            \@bmpsize@read{#1}{2}{\bmpsize@off}%
605            \@bmpsize@num@two\bmpsize@entries
```

```
606              \edef\bmpsize@off{\the\numexpr\bmpsize@off+2}%
607              \@bmpsize@loop{%
608                \ifnum\bmpsize@entries=0\relax
609                  \expandafter\@firstofone
610                \else
611                  \expandafter\@gobble
612                \fi
613                {%
614                  \@bmpsize@break
615                }%
616                \edef\bmpsize@entries{\the\numexpr\bmpsize@entries-1}%
617                \@bmpsize@read{#1}{12}{\bmpsize@off}%
618                \edef\bmpsize@off{\the\numexpr\bmpsize@off+12}%
619                \@bmpsize@num@two\bmpsize@tag
620                \ifnum\bmpsize@tag=296\relax
621                  \expandafter\@firstofone
622                \else
623                  \expandafter\@gobble
624                \fi
625                {%
626                  \@bmpsize@skip@four
627                  \@bmpsize@skip@two
628                  \@bmpsize@num@two\bmpsize@temp
629                  \ifcase\bmpsize@temp\relax
630                  \or
631                    \let\bmpsize@unit\relax
632                  \or
633                    \def\bmpsize@unit{1in}%
634                  \or
635                    \def\bmpsize@unit{1cm}%
636                  \else
637                    \let\bmpsize@unit\relax
638                  \fi
639                  \ifcase\bmpsize@temp\relax
640                  \or
641                  \or
642                    \def\bmpsize@exifdensity{1}%
643                  \or
644                    \def\bmpsize@exifdensity{1}%
645                  \else
646                    \let\bmpsize@exifdensity\bmpsize@exifdensity
647                  \fi
648                }%
649                \ifnum\bmpsize@tag=282\relax
650                  \expandafter\@firstofone
651                \else
652                  \expandafter\@gobble
653                \fi
654                {%
655                  \@bmpsize@skip@four
656                  \@bmpsize@skip@two
657                  \@bmpsize@num@four\bmpsize@temp
658                  \@bmpsize@read{#1}{8}{\numexpr\bmpsize@temp+\bmpsize@exifoffset\relax}%
659                  \@bmpsize@num@four\bmpsize@pixelx
660                  \@bmpsize@num@four\bmpsize@temp
661                  \ifnum\bmpsize@temp=1\relax
662                    \expandafter\@gobble
663                  \else
664                    \expandafter\@firstofone
665                  \fi
666                  {%
667                    \edef\bmpsize@pixelxdenom{\the\numexpr\bmpsize@temp}%
```

18

```
668              }%
669            }%
670            \ifnum\bmpsize@tag=283\relax
671              \expandafter\@firstofone
672            \else
673              \expandafter\@gobble
674            \fi
675            {%
676              \@bmpsize@skip@four
677              \@bmpsize@skip@two
678              \@bmpsize@num@four\bmpsize@temp
679              \@bmpsize@read{#1}{8}{\numexpr\bmpsize@temp+\bmpsize@exifoffset\relax}%
680              \@bmpsize@num@four\bmpsize@pixely
681              \@bmpsize@num@four\bmpsize@temp
682              \ifnum\bmpsize@temp=1\relax
683                \expandafter\@gobble
684              \else
685                \expandafter\@firstofone
686              \fi
687              {%
688                \edef\bmpsize@pixelydenom{\the\numexpr\bmpsize@temp}%
689              }%
690            }%
691          }%
692          \@bmpsize@bigendiantrue
693        }%
694      }%
695    }{%
696      \edef\bmpsize@temp{\the\numexpr\bmpsize@temp-192}%
697      \ifcase\bmpsize@temp\relax
698      \or
699      \or
700      \or
701      \or
702        \def\bmpsize@temp{-1}%
703      \or
704      \or
705      \or
706      \or
707        \def\bmpsize@temp{-1}%
708      \or
709      \or
710      \or
711      \or
712        \def\bmpsize@temp{-1}%
713      \or
714      \or
715      \or
716      \else
717        \def\bmpsize@temp{-1}%
718      \fi
719      \ifnum\bmpsize@temp=-1\relax
720        \expandafter\@gobble
721      \else
722        \expandafter\@firstofone
723      \fi
724      {%
725        \@bmpsize@read{#1}{4}{\numexpr\bmpsize@offset+5\relax}%
726        \@bmpsize@num@two\bmpsize@pixelheight
727        \@bmpsize@num@two\bmpsize@pixelwidth
728        \ifnum\bmpsize@pixelheight=0\relax
729          \expandafter\@firstofone
```

```
730            \else
731              \expandafter\@gobble
732            \fi
733            {%
734              \let\bmpsize@pixelheight\relax
735              \@bmpsize@stop
736            }%
737            \@bmpsize@ok
738            \@bmpsize@stop
739          }%
740          \@bmpsize@num@two\bmpsize@length
741        }%
742      }%
743      \edef\bmpsize@offset{\the\numexpr\bmpsize@offset+\bmpsize@length+2}%
744    }%
745    \@bmpsize@stop
746    \@nil
747    \@bmpsize@end
748 }%
749 ⟨/base⟩
```

2.2.3 bmp

¡*ignore¿

```
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
```

¡/ignore¿

\bmpsize@read@bmp

```
750 ⟨*base⟩
751 \def\bmpsize@read@bmp#1{%
752   \@bmpsize@init
753   \@bmpsize@bigendianfalse
754   \@bmpsize@read{#1}{26}{0}%
755   \@bmpsize@grab\bmpsize@temp{2}%
756   \@bmpsize@skip@two
757   \ifnum\pdfstrcmp{\bmpsize@temp}{424D}=\z@
758   \else
759     \expandafter\@bmpsize@stop
760   \fi
761   \@bmpsize@skip@four
762   \@bmpsize@skip@four
763   \@bmpsize@skip@four
764   \@bmpsize@num@two\bmpsize@temp
765   \ifnum\bmpsize@temp=12\relax
766     \expandafter\@firstofone
767   \else
768     \expandafter\@gobble
769   \fi
770   {%
771     \@bmpsize@skip@two
772     \@bmpsize@num@two\bmpsize@pixelwidth
773     \@bmpsize@num@two\bmpsize@pixelheight
774     \@bmpsize@ok
775     \@bmpsize@stop
776   }%
777   \ifnum\bmpsize@temp=64\relax
778     \expandafter\@firstofone
779   \else
780     \expandafter\@gobble
781   \fi
782   {%
783     \@bmpsize@skip@two
784     \@bmpsize@num@two\bmpsize@pixelwidth
785     \@bmpsize@num@two\bmpsize@pixelheight
786     \@bmpsize@ok
787     \@bmpsize@stop
788   }%
789   \@bmpsize@skip@two
790   \@bmpsize@num@four\bmpsize@pixelwidth
791   \@bmpsize@absnumtrue
792   \@bmpsize@num@four\bmpsize@pixelheight
793   \@bmpsize@absnumfalse
794   \@bmpsize@ok
795   \@bmpsize@read{#1}{8}{38}%
796   \@bmpsize@num@four\bmpsize@pixelx
797   \@bmpsize@num@four\bmpsize@pixely
798   \def\bmpsize@unit{100cm}%
799   \@bmpsize@stop
800   \@nil
801   \@bmpsize@end
802 }%
803 ⟨/base⟩
```

2.2.4   gif

¡*ignore¿

```
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
```

¡/ignore¿

\bmpsize@read@gif

```
804 ⟨∗base⟩
805 \def\bmpsize@read@gif#1{%
806   \@bmpsize@init
807   \@bmpsize@bigendianfalse
808   \@bmpsize@read{#1}{13}{0}%
809   \@bmpsize@grab\bmpsize@temp{3}%
810   \@bmpsize@skip@two
811   \@bmpsize@skip@one
812   \ifnum\pdfstrcmp{\bmpsize@temp}{474946}=\z@
813   \else
814     \expandafter\@bmpsize@stop
815   \fi
816   \@bmpsize@skip@two
817   \@bmpsize@skip@one
818   \@bmpsize@num@two\bmpsize@pixelwidth
819   \@bmpsize@num@two\bmpsize@pixelheight
820   \@bmpsize@skip@two
821   \@bmpsize@num@one\bmpsize@temp
822   \ifnum\bmpsize@temp=0\relax
823     \expandafter\@gobble
824   \else
825     \expandafter\@firstofone
826   \fi
827   {%
828     \edef\bmpsize@pixelx{\the\numexpr\bmpsize@temp+15}%
829     \def\bmpsize@pixely{64}%
830   }%
831   \@bmpsize@ok
832   \@bmpsize@stop
833   \@nil
834   \@bmpsize@end
835 }%
836 ⟨/base⟩
```

2.2.5 tiff

¡*ignore¿

```
begin tiff
% defaults
assign {1in} -> $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
  % 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 {1in} -> $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
```

¡/ignore¿

\bmpsize@read@tiff

```
837 ⟨∗base⟩
838 \def\bmpsize@read@tiff#1{%
839   \@bmpsize@init
840   \def\bmpsize@unit{1in}%
841   \@bmpsize@read{#1}{8}{0}%
842   \@bmpsize@grab\bmpsize@temp{2}%
843   \@bmpsize@skip@two
844   \ifnum\pdfstrcmp{\bmpsize@temp}{4949}=\z@
845     \expandafter\@firstoftwo
846   \else
847     \expandafter\@secondoftwo
848   \fi
849   {%
850     \@bmpsize@bigendianfalse
851   }{%
852     \ifnum\pdfstrcmp{\bmpsize@temp}{4D4D}=\z@
853     \else
854       \expandafter\@bmpsize@stop
855     \fi
856     \@bmpsize@bigendiantrue
857   }%
858   \@bmpsize@num@two\bmpsize@temp
859   \ifnum\bmpsize@temp=42\relax
860   \else
861     \expandafter\@bmpsize@stop
862   \fi
863   \@bmpsize@num@four\bmpsize@offset
864   \@bmpsize@read{#1}{2}{\bmpsize@offset}%
865   \edef\bmpsize@offset{\the\numexpr\bmpsize@offset+2}%
866   \@bmpsize@num@two\bmpsize@entries
867   \@bmpsize@ok
868   \@bmpsize@loop{%
869     \ifnum\bmpsize@entries=0\relax
870       \expandafter\@firstofone
871     \else
872       \expandafter\@gobble
873     \fi
874     {%
875       \@bmpsize@stop
876     }%
877     \edef\bmpsize@entries{\the\numexpr\bmpsize@entries-1}%
```

```
878    \@bmpsize@read{#1}{12}{\bmpsize@offset}%
879    \edef\bmpsize@offset{\the\numexpr\bmpsize@offset+12}%
880    \@bmpsize@num@two\bmpsize@tag
881    \ifnum\bmpsize@temp=296\relax
882      \expandafter\@firstofone
883    \else
884      \expandafter\@gobble
885    \fi
886    {%
887      \@bmpsize@skip@four
888      \@bmpsize@skip@two
889      \@bmpsize@num@two\bmpsize@temp
890      \ifcase\bmpsize@temp\relax
891      \or
892        \let\bmpsize@unit\relax
893      \or
894        \def\bmpsize@unit{1in}%
895      \or
896        \def\bmpsize@unit{1cm}%
897      \else
898        \let\bmpsize@unit\relax
899      \fi
900    }%
901    \ifnum\bmpsize@tag=256\relax
902      \expandafter\@firstofone
903    \else
904      \expandafter\@gobble
905    \fi
906    {%
907      \@bmpsize@skip@four
908      \@bmpsize@skip@two
909      \@bmpsize@num@four\bmpsize@pixelwidth
910    }%
911    \ifnum\bmpsize@tag=257\relax
912      \expandafter\@firstofone
913    \else
914      \expandafter\@gobble
915    \fi
916    {%
917      \@bmpsize@skip@four
918      \@bmpsize@skip@two
919      \@bmpsize@num@four\bmpsize@pixelheight
920    }%
921    \ifnum\bmpsize@tag=282\relax
922      \expandafter\@firstofone
923    \else
924      \expandafter\@gobble
925    \fi
926    {%
927      \@bmpsize@skip@four
928      \@bmpsize@skip@two
929      \@bmpsize@num@four\bmpsize@temp
930      \@bmpsize@read{#1}{8}{\bmpsize@temp}%
931      \@bmpsize@num@four\bmpsize@pixelx
932      \@bmpsize@num@four\bmpsize@temp
933      \ifnum\bmpsize@temp=1\relax
934        \expandafter\@gobble
935      \else
936        \expandafter\@firstofone
937      \fi
938      {%
939        \edef\bmpsize@pixelxdenom{\the\numexpr\bmpsize@temp}%
```

```
940        }%
941      }%
942      \ifnum\bmpsize@tag=283\relax
943        \expandafter\@firstofone
944      \else
945        \expandafter\@gobble
946      \fi
947      {%
948        \@bmpsize@skip@four
949        \@bmpsize@skip@two
950        \@bmpsize@num@four\bmpsize@temp
951        \@bmpsize@read{#1}{8}{\bmpsize@temp}%
952        \@bmpsize@num@four\bmpsize@pixely
953        \@bmpsize@num@four\bmpsize@temp
954        \ifnum\bmpsize@temp=1\relax
955          \expandafter\@gobble
956        \else
957          \expandafter\@firstofone
958        \fi
959        {%
960          \edef\bmpsize@pixelydenom{\the\numexpr\bmpsize@temp}%
961        }%
962      }%
963    }%
964    \@bmpsize@stop
965    \@nil
966    \@bmpsize@end
967 }%
968 ⟨/base⟩
```

2.2.6   pnm

¡*ignore¿

```
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
  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
```

¡/ignore¿

\bmpsize@read@pnm

```
969 ⟨∗base⟩
970 \def\bmpsize@read@pnm#1{%
971   \@bmpsize@init
972   \def\bmpsize@offset{0}%
973   \@bmpsize@read{#1}{3}{\bmpsize@offset}%
974   \def\bmpsize@offset{3}%
975   \@bmpsize@grab\bmpsize@temp{1}%
```

```
976    \@bmpsize@skip@one
977    \ifnum\pdfstrcmp{\bmpsize@temp}{50}=\z@
978    \else
979      \expandafter\@bmpsize@stop
980    \fi
981    \@bmpsize@grab\bmpsize@temp{1}%
982    \@bmpsize@skip@one
983    \ifnum\pdfstrcmp{\bmpsize@temp}{31}<\z@
984      \expandafter\@bmpsize@stop
985    \fi
986    \ifnum\pdfstrcmp{\bmpsize@temp}{36}>\z@
987      \expandafter\@bmpsize@stop
988    \fi
989    \@bmpsize@grab\bmpsize@temp{1}%
990    \@bmpsize@skip@one
991    \ifcase 0\@bmpsize@iswhite\bmpsize@temp
992      \expandafter\@gobble
993    \else
994      \expandafter\@firstofone
995    \fi
996    {%
997      \@bmpsize@stop
998    }%
999    \@bmpsize@loop{%
1000      \@bmpsize@fillbuf{#1}%
1001      \@bmpsize@grab\bmpsize@temp{1}%
1002      \@bmpsize@skip@one
1003      \ifcase 0\@bmpsize@iswhite\bmpsize@temp
1004        \expandafter\@gobble
1005      \else
1006        \expandafter\@firstofone
1007      \fi
1008      {%
1009        \ifnum\pdfstrcmp{\bmpsize@temp}{23}=\z@
1010          \expandafter\@firstoftwo
1011        \else
1012          \expandafter\@secondoftwo
1013        \fi
1014        {%
1015          \@bmpsize@loop{%
1016            \@bmpsize@fillbuf{#1}%
1017            \@bmpsize@grab\bmpsize@temp{1}%
1018            \@bmpsize@skip@one
1019            \ifnum\pdfstrcmp{\bmpsize@temp}{0A}=\z@
1020              \expandafter\@firstoftwo
1021            \else
1022              \expandafter\@secondoftwo
1023            \fi
1024            {%
1025              \@bmpsize@break
1026            }{%
1027              \ifnum\pdfstrcmp{\bmpsize@temp}{0D}=\z@
1028                \expandafter\@firstofone
1029              \else
1030                \expandafter\@gobble
1031              \fi
1032              {%
1033                \@bmpsize@break
1034              }%
1035            }%
1036          }%
1037        }{%
```

```
1038        \@bmpsize@pushback\bmpsize@temp
1039        \@bmpsize@break
1040      }%
1041    }%
1042  }%
1043  \def\bmpsize@tempnum{}%
1044  \@bmpsize@loop{%
1045    \@bmpsize@fillbuf{#1}%
1046    \@bmpsize@grab\bmpsize@temp{1}%
1047    \@bmpsize@skip@one
1048    \ifcase 0\@bmpsize@isdigit\bmpsize@temp
1049      \expandafter\@firstoftwo
1050    \else
1051      \expandafter\@secondoftwo
1052    \fi
1053    {%
1054      \@bmpsize@append\bmpsize@tempnum\bmpsize@tempnum\bmpsize@temp
1055    }{%
1056      \ifcase 0\@bmpsize@iswhite\bmpsize@temp
1057        \expandafter\@firstoftwo
1058      \else
1059        \expandafter\@secondoftwo
1060      \fi
1061      {%
1062        \@bmpsize@break
1063      }{%
1064        \@bmpsize@stop
1065      }%
1066    }%
1067  }%
1068  \edef\bmpsize@pixelwidth{\pdfunescapehex{\bmpsize@tempnum}}%
1069  \@bmpsize@loop{%
1070    \@bmpsize@fillbuf{#1}%
1071    \@bmpsize@grab\bmpsize@temp{1}%
1072    \@bmpsize@skip@one
1073    \ifcase 0\@bmpsize@iswhite\bmpsize@temp
1074      \expandafter\@gobble
1075    \else
1076      \expandafter\@firstofone
1077    \fi
1078    {%
1079      \@bmpsize@pushback\bmpsize@temp
1080      \@bmpsize@break
1081    }%
1082  }%
1083  \def\bmpsize@tempnum{}%
1084  \@bmpsize@loop{%
1085    \@bmpsize@fillbuf{#1}%
1086    \@bmpsize@grab\bmpsize@temp{1}%
1087    \@bmpsize@skip@one
1088    \ifcase 0\@bmpsize@isdigit\bmpsize@temp
1089      \expandafter\@firstoftwo
1090    \else
1091      \expandafter\@secondoftwo
1092    \fi
1093    {%
1094      \@bmpsize@append\bmpsize@tempnum\bmpsize@tempnum\bmpsize@temp
1095    }{%
1096      \ifcase 0\@bmpsize@iswhite\bmpsize@temp
1097        \expandafter\@firstoftwo
1098      \else
1099        \expandafter\@secondoftwo
```

```
1100      \fi
1101      {%
1102        \@bmpsize@break
1103      }{%
1104        \@bmpsize@stop
1105      }%
1106    }%
1107  }%
1108  \edef\bmpsize@pixelheight{\pdfunescapehex{\bmpsize@tempnum}}%
1109  \@bmpsize@ok
1110  \@bmpsize@stop
1111  \@nil
1112  \@bmpsize@end
1113 }%
1114 ⟨/base⟩
```

2.2.7   pam

¡*ignore¿

```
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
repeat
end
```

¡/ignore¿

\bmpsize@read@pam

```
1115 ⟨∗base⟩
1116 \def\bmpsize@read@pam#1{%
1117   \@bmpsize@init
1118   \@bmpsize@read{#1}{3}{0}%
1119   \def\bmpsize@offset{3}%
1120   \let\bmpsize@off\bmpsize@offset
1121   \@bmpsize@grab\bmpsize@temp{3}%
1122   \@bmpsize@skip@two
1123   \@bmpsize@skip@one
1124   \ifnum\pdfstrcmp{\bmpsize@temp}{50370A}=\z@
1125   \else
1126     \expandafter\@bmpsize@stop
1127   \fi
1128   \@bmpsize@loop{%
1129     \@bmpsize@fillbuf{#1}%
1130     \@bmpsize@grab\bmpsize@temp{1}%
1131     \@bmpsize@skip@one
1132     \ifcase 0\@bmpsize@iswhite\bmpsize@temp
1133       \expandafter\@firstoftwo
1134     \else
1135       \expandafter\@secondoftwo
1136     \fi
1137     {%
1138       \edef\bmpsize@off{\the\numexpr\bmpsize@off+1}%
1139     }{%
1140       \ifnum\pdfstrcmp{\bmpsize@temp}{23}=\z@
1141         \expandafter\@firstoftwo
1142       \else
```

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

```
1205            \expandafter\@firstoftwo
1206          \else
1207            \expandafter\@secondoftwo
1208          \fi
1209          {%
1210            \edef\bmpsize@off{\the\numexpr\bmpsize@off+1}%
1211            \@bmpsize@append\bmpsize@tempnum\bmpsize@tempnum\bmpsize@temp
1212          }{%
1213            \@bmpsize@pushback\bmpsize@temp
1214            \@bmpsize@break
1215          }%
1216        }%
1217        \@bmpsize@loop{%
1218          \@bmpsize@fillbuf{#1}%
1219          \@bmpsize@grab\bmpsize@temp{1}%
1220          \@bmpsize@skip@one
1221          \edef\bmpsize@off{\the\numexpr\bmpsize@off+1}%
1222          \ifnum\pdfstrcmp{\bmpsize@temp}{0A}=\z@
1223            \expandafter\@firstofone
1224          \else
1225            \expandafter\@gobble
1226          \fi
1227          {%
1228            \@bmpsize@break
1229          }%
1230        }%
1231        \edef\bmpsize@pixelwidth{\pdfunescapehex{\bmpsize@tempnum}}%
1232      }{%
1233        \@bmpsize@grab\bmpsize@temp{1}%
1234        \@bmpsize@skip@one
1235        \@bmpsize@append\bmpsize@head\bmpsize@head\bmpsize@temp
1236        \ifnum\pdfstrcmp{\bmpsize@head}{454E44484452}=\z@
1237          \expandafter\@firstoftwo
1238        \else
1239          \expandafter\@secondoftwo
1240        \fi
1241        {%
1242          \@bmpsize@ok
1243          \@bmpsize@stop
1244        }{%
1245          \ifnum\pdfstrcmp{\bmpsize@head}{484549474854}=\z@
1246            \expandafter\@firstoftwo
1247          \else
1248            \expandafter\@secondoftwo
1249          \fi
1250          {%
1251            \edef\bmpsize@off{\the\numexpr\bmpsize@off+6}%
1252            \@bmpsize@loop{%
1253              \@bmpsize@fillbuf{#1}%
1254              \@bmpsize@grab\bmpsize@temp{1}%
1255              \@bmpsize@skip@one
1256              \ifcase 0\@bmpsize@iswhite\bmpsize@temp
1257                \expandafter\@firstoftwo
1258              \else
1259                \expandafter\@secondoftwo
1260              \fi
1261              {%
1262                \edef\bmpsize@off{\the\numexpr\bmpsize@off+1}%
1263              }{%
1264                \ifcase 0\@bmpsize@isdigit\bmpsize@temp
1265                  \expandafter\@firstoftwo
1266                \else
```

```
1267                    \expandafter\@secondoftwo
1268                  \fi
1269                  {%
1270                    \edef\bmpsize@off{\the\numexpr\bmpsize@off+1}%
1271                    \@bmpsize@break
1272                  }{%
1273                    \@bmpsize@stop
1274                  }%
1275                }%
1276              }%
1277              \let\bmpsize@tempnum\bmpsize@temp
1278              \@bmpsize@loop{%
1279                \@bmpsize@fillbuf{#1}%
1280                \@bmpsize@grab\bmpsize@temp{1}%
1281                \@bmpsize@skip@one
1282                \ifcase 0\@bmpsize@isdigit\bmpsize@temp
1283                  \expandafter\@firstoftwo
1284                \else
1285                  \expandafter\@secondoftwo
1286                \fi
1287                {%
1288                  \edef\bmpsize@off{\the\numexpr\bmpsize@off+1}%
1289                  \@bmpsize@append\bmpsize@tempnum\bmpsize@tempnum\bmpsize@temp
1290                }{%
1291                  \@bmpsize@pushback\bmpsize@temp
1292                  \@bmpsize@break
1293                }%
1294              }%
1295              \@bmpsize@loop{%
1296                \@bmpsize@fillbuf{#1}%
1297                \@bmpsize@grab\bmpsize@temp{1}%
1298                \@bmpsize@skip@one
1299                \edef\bmpsize@off{\the\numexpr\bmpsize@off+1}%
1300                \ifnum\pdfstrcmp{\bmpsize@temp}{0A}=\z@
1301                  \expandafter\@firstofone
1302                \else
1303                  \expandafter\@gobble
1304                \fi
1305                {%
1306                  \@bmpsize@break
1307                }%
1308              }%
1309              \edef\bmpsize@pixelheight{\pdfunescapehex{\bmpsize@tempnum}}%
1310            }{%
1311              \@bmpsize@pushback\bmpsize@head
1312              \@bmpsize@loop{%
1313                \@bmpsize@fillbuf{#1}%
1314                \@bmpsize@grab\bmpsize@temp{1}%
1315                \@bmpsize@skip@one
1316                \edef\bmpsize@off{\the\numexpr\bmpsize@off+1}%
1317                \ifnum\pdfstrcmp{\bmpsize@temp}{0A}=\z@
1318                  \expandafter\@firstofone
1319                \else
1320                  \expandafter\@gobble
1321                \fi
1322                {%
1323                  \@bmpsize@break
1324                }%
1325              }%
1326            }%
1327          }%
1328        }%
```

```
1329          }%
1330        }%
1331      }%
1332    \@bmpsize@stop
1333    \@nil
1334    \@bmpsize@end
1335 }%
1336 ⟨/base⟩
```

2.2.8   xpm

¡*ignore¿

```
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 ["/"]
              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
```

¡/ignore¿

\bmpsize@read@xpm

```
1337 ⟨∗base⟩
1338 \def\bmpsize@read@xpm#1{%
1339   \@bmpsize@init
1340   \@bmpsize@read{#1}{9}{0}%
1341   \@bmpsize@grab\bmpsize@temp{9}%
1342   \@bmpsize@skip@four
1343   \@bmpsize@skip@four
1344   \@bmpsize@skip@one
1345   \def\bmpsize@offset{9}%
1346   \ifnum\pdfstrcmp{\bmpsize@temp}{2F2A2058504D202A2F}=\z@
```

```
1347  \else
1348    \expandafter\@bmpsize@stop
1349  \fi
1350  \@bmpsize@loop{%
1351    \@bmpsize@fillbuf{#1}%
1352    \@bmpsize@grab\bmpsize@temp{1}%
1353    \@bmpsize@skip@one
1354    \ifnum\pdfstrcmp{\bmpsize@temp}{22}=\z@
1355      \expandafter\@firstofone
1356    \else
1357      \expandafter\@gobble
1358    \fi
1359    {%
1360      \@bmpsize@break
1361    }%
1362    \ifnum\pdfstrcmp{\bmpsize@temp}{2F}=\z@
1363      \expandafter\@firstofone
1364    \else
1365      \expandafter\@gobble
1366    \fi
1367    {%
1368      \@bmpsize@fillbuf{#1}%
1369      \@bmpsize@grab\bmpsize@temp{1}%
1370      \@bmpsize@skip@one
1371      \ifnum\pdfstrcmp{\bmpsize@temp}{2A}=\z@
1372        \expandafter\@firstofone
1373      \else
1374        \expandafter\@gobble
1375      \fi
1376      {%
1377        \@bmpsize@loop{%
1378          \@bmpsize@fillbuf{#1}%
1379          \@bmpsize@grab\bmpsize@temp{1}%
1380          \@bmpsize@skip@one
1381          \ifnum\pdfstrcmp{\bmpsize@temp}{2A}=\z@
1382            \expandafter\@firstofone
1383          \else
1384            \expandafter\@gobble
1385          \fi
1386          {%
1387            \@bmpsize@loop{%
1388              \@bmpsize@fillbuf{#1}%
1389              \@bmpsize@grab\bmpsize@temp{1}%
1390              \@bmpsize@skip@one
1391              \ifnum\pdfstrcmp{\bmpsize@temp}{2F}=\z@
1392                \expandafter\@firstofone
1393              \else
1394                \expandafter\@gobble
1395              \fi
1396              {%
1397                \@bmpsize@break
1398              }%
1399              \ifnum\pdfstrcmp{\bmpsize@temp}{2A}=\z@
1400                \expandafter\@gobble
1401              \else
1402                \expandafter\@firstofone
1403              \fi
1404              {%
1405                \@bmpsize@break
1406              }%
1407            }%
1408            \ifnum\pdfstrcmp{\bmpsize@temp}{2F}=\z@
```

```
1409        \expandafter\@firstofone
1410      \else
1411        \expandafter\@gobble
1412      \fi
1413      {%
1414        \@bmpsize@break
1415      }%
1416    }%
1417    }%
1418    }%
1419    }%
1420  }%
1421  \def\bmpsize@tempnum{}%
1422  \@bmpsize@loop{%
1423    \@bmpsize@fillbuf{#1}%
1424    \@bmpsize@grab\bmpsize@temp{1}%
1425    \@bmpsize@skip@one
1426    \ifcase 0\@bmpsize@iswhite\bmpsize@temp
1427      \expandafter\@gobble
1428    \else
1429      \expandafter\@firstofone
1430    \fi
1431    {%
1432      \ifcase 0\@bmpsize@isdigit\bmpsize@temp
1433        \expandafter\@firstoftwo
1434      \else
1435        \expandafter\@secondoftwo
1436      \fi
1437      {%
1438        \@bmpsize@append\bmpsize@tempnum\bmpsize@tempnum\bmpsize@temp
1439        \@bmpsize@break
1440      }{%
1441        \@bmpsize@stop
1442      }%
1443    }%
1444  }%
1445  \@bmpsize@loop{%
1446    \@bmpsize@fillbuf{#1}%
1447    \@bmpsize@grab\bmpsize@temp{1}%
1448    \@bmpsize@skip@one
1449    \ifcase 0\@bmpsize@isdigit\bmpsize@temp
1450      \expandafter\@firstoftwo
1451    \else
1452      \expandafter\@secondoftwo
1453    \fi
1454    {%
1455      \@bmpsize@append\bmpsize@tempnum\bmpsize@tempnum\bmpsize@temp
1456    }{%
1457      \ifcase 0\@bmpsize@iswhite\bmpsize@temp
1458        \expandafter\@firstoftwo
1459      \else
1460        \expandafter\@secondoftwo
1461      \fi
1462      {%
1463        \@bmpsize@break
1464      }{%
1465        \@bmpsize@stop
1466      }%
1467    }%
1468  }%
1469  \edef\bmpsize@pixelwidth{\pdfunescapehex{\bmpsize@tempnum}}%
1470  \def\bmpsize@tempnum{}%
```

```
1471  \@bmpsize@loop{%
1472    \@bmpsize@fillbuf{#1}%
1473    \@bmpsize@grab\bmpsize@temp{1}%
1474    \@bmpsize@skip@one
1475    \ifcase 0\@bmpsize@iswhite\bmpsize@temp
1476      \expandafter\@gobble
1477    \else
1478      \expandafter\@firstofone
1479    \fi
1480    {%
1481      \ifcase 0\@bmpsize@isdigit\bmpsize@temp
1482        \expandafter\@firstoftwo
1483      \else
1484        \expandafter\@secondoftwo
1485      \fi
1486      {%
1487        \@bmpsize@append\bmpsize@tempnum\bmpsize@tempnum\bmpsize@temp
1488        \@bmpsize@break
1489      }{%
1490        \@bmpsize@stop
1491      }%
1492    }%
1493  }%
1494  \@bmpsize@loop{%
1495    \@bmpsize@fillbuf{#1}%
1496    \@bmpsize@grab\bmpsize@temp{1}%
1497    \@bmpsize@skip@one
1498    \ifcase 0\@bmpsize@isdigit\bmpsize@temp
1499      \expandafter\@firstoftwo
1500    \else
1501      \expandafter\@secondoftwo
1502    \fi
1503    {%
1504      \@bmpsize@append\bmpsize@tempnum\bmpsize@tempnum\bmpsize@temp
1505    }{%
1506      \ifcase 0\@bmpsize@iswhite\bmpsize@temp
1507        \expandafter\@firstoftwo
1508      \else
1509        \expandafter\@secondoftwo
1510      \fi
1511      {%
1512        \@bmpsize@break
1513      }{%
1514        \@bmpsize@stop
1515      }%
1516    }%
1517  }%
1518  \edef\bmpsize@pixelheight{\pdfunescapehex{\bmpsize@tempnum}}%
1519  \@bmpsize@ok
1520  \@bmpsize@stop
1521  \@nil
1522  \@bmpsize@end
1523 }%
1524 ⟨/base⟩
```

2.2.9 tga

¡*ignore¿

```
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
```

¡/ignore¿

\bmpsize@read@tga

```
1525 ⟨∗base⟩
1526 \def\bmpsize@read@tga#1{%
1527   \@bmpsize@init
1528   \@bmpsize@bigendianfalse
1529   \@bmpsize@read{#1}{16}{1}%
1530   \@bmpsize@grab\bmpsize@temp{1}%
1531   \@bmpsize@skip@one
1532   \ifnum\pdfstrcmp{\bmpsize@temp}{00}=\z@
1533     \expandafter\@gobble
1534   \else
1535     \expandafter\@firstofone
1536   \fi
1537   {%
1538     \ifnum\pdfstrcmp{\bmpsize@temp}{01}=\z@
1539       \expandafter\@gobble
1540     \else
1541       \expandafter\@firstofone
1542     \fi
1543     {%
1544       \@bmpsize@stop
1545     }%
1546   }%
1547   \@bmpsize@skip@four
```

```
1548    \@bmpsize@skip@four
1549    \@bmpsize@skip@two
1550    \@bmpsize@num@two\bmpsize@pixelwidth
1551    \@bmpsize@num@two\bmpsize@pixelheight
1552    \@bmpsize@ok
1553    \@bmpsize@size{#1}{26}\bmpsize@temp  \@bmpsize@read{#1}{26}{\numexpr\bmpsize@temp-26\relax
1554    \@bmpsize@num@four\bmpsize@offset
1555    \@bmpsize@skip@four
1556    \@bmpsize@grab\bmpsize@temp{18}%
1557    \@bmpsize@skip@four
1558    \@bmpsize@skip@four
1559    \@bmpsize@skip@four
1560    \@bmpsize@skip@four
1561    \@bmpsize@skip@two
1562    \ifnum\pdfstrcmp{\bmpsize@temp}{54525545564953494F4E2D5846494C452E00}=\z@
1563      \expandafter\@gobble
1564    \else
1565      \expandafter\@firstofone
1566    \fi
1567    {%
1568      \@bmpsize@stop
1569    }%
1570    \ifnum\bmpsize@offset=0\relax
1571      \expandafter\@firstofone
1572    \else
1573      \expandafter\@gobble
1574    \fi
1575    {%
1576      \@bmpsize@stop
1577    }%
1578    \@bmpsize@read{#1}{4}{\numexpr\bmpsize@offset+474\relax}%
1579    \@bmpsize@num@two\bmpsize@pixelx
1580    \@bmpsize@num@two\bmpsize@pixely
1581    \ifnum\bmpsize@pixely=0\relax
1582      \expandafter\@firstofone
1583    \else
1584      \expandafter\@gobble
1585    \fi
1586    {%
1587      \let\bmpsize@pixelx\relax
1588      \let\bmpsize@pixely\relax
1589    }%
1590    \@bmpsize@stop
1591    \@nil
1592    \@bmpsize@end
1593 }%
1594 ⟨/base⟩
```

2.2.10 pcx

¡*ignore¿

```
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
```

42

\bmpsize@read@pcx

```
1595 ⟨∗base⟩
1596 \def\bmpsize@read@pcx#1{%
1597   \@bmpsize@init
1598   \@bmpsize@bigendianfalse
1599   \@bmpsize@read{#1}{16}{0}%
1600   \@bmpsize@grab\bmpsize@temp{1}%
1601   \@bmpsize@skip@one
1602   \ifnum\pdfstrcmp{\bmpsize@temp}{0A}=\z@
1603   \else
1604     \expandafter\@bmpsize@stop
1605   \fi
1606   \@bmpsize@skip@one
1607   \@bmpsize@num@one\bmpsize@temp
1608   \ifnum\bmpsize@temp=1\relax
1609   \else
1610     \expandafter\@bmpsize@stop
1611   \fi
1612   \@bmpsize@skip@one
1613   \@bmpsize@num@two\bmpsize@pixelwidth
1614   \@bmpsize@num@two\bmpsize@pixelheight
1615   \@bmpsize@num@two\bmpsize@temp
1616   \edef\bmpsize@pixelwidth{\the\numexpr\bmpsize@temp-\bmpsize@pixelwidth+1}%
1617   \@bmpsize@num@two\bmpsize@temp
1618   \edef\bmpsize@pixelheight{\the\numexpr\bmpsize@temp-\bmpsize@pixelheight+1}%
1619   \ifnum\bmpsize@pixelwidth>0\relax
1620   \else
1621     \expandafter\@bmpsize@stop
1622   \fi
1623   \ifnum\bmpsize@pixelheight>0\relax
1624   \else
1625     \expandafter\@bmpsize@stop
1626   \fi
1627   \@bmpsize@ok
1628   \@bmpsize@num@two\bmpsize@pixelx
1629   \@bmpsize@num@two\bmpsize@pixely
1630   \def\bmpsize@unit{1in}%
1631   \@bmpsize@stop
1632   \@nil
1633   \@bmpsize@end
1634 }%
1635 ⟨/base⟩
```

2.2.11 msp

¡*ignore¿

```
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 {1in} -> $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
```

¡/ignore¿

\bmpsize@read@msp

```
1636 ⟨∗base⟩
1637 \def\bmpsize@read@msp#1{%
1638   \@bmpsize@init
1639   \@bmpsize@bigendianfalse
1640   \@bmpsize@read{#1}{16}{0}%
1641   \@bmpsize@grab\bmpsize@temp{4}%
1642   \@bmpsize@skip@four
1643   \ifnum\pdfstrcmp{\bmpsize@temp}{44616E4D}=\z@
1644     \expandafter\@gobble
1645   \else
1646     \expandafter\@firstofone
1647   \fi
1648   {%
1649     \ifnum\pdfstrcmp{\bmpsize@temp}{4C696E53}=\z@
1650     \else
1651       \expandafter\@bmpsize@stop
1652     \fi
1653   }%
1654   \@bmpsize@num@two\bmpsize@pixelwidth
1655   \@bmpsize@num@two\bmpsize@pixelheight
1656   \@bmpsize@ok
1657   \@bmpsize@num@two\bmpsize@pixelx
1658   \@bmpsize@num@two\bmpsize@pixely
1659   \def\bmpsize@unit{1in}%
1660   \ifnum\bmpsize@pixelx=0\relax
1661     \expandafter\@firstofone
1662   \else
1663     \expandafter\@gobble
1664   \fi
1665   {%
1666     \@bmpsize@num@two\bmpsize@pixelx
1667     \@bmpsize@num@two\bmpsize@pixely
1668   }%
1669   \@bmpsize@stop
1670   \@nil
1671   \@bmpsize@end
1672 }%
```

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

```
1674 ⟨*base⟩
1675 \def\bmpsize@read@sgi#1{%
1676   \@bmpsize@init
1677   \@bmpsize@bigendiantrue
1678   \@bmpsize@read{#1}{10}{0}%
1679   \@bmpsize@grab\bmpsize@temp{2}%
1680   \@bmpsize@skip@two
1681   \ifnum\pdfstrcmp{\bmpsize@temp}{01DA}=\z@
1682   \else
1683     \expandafter\@bmpsize@stop
1684   \fi
1685   \@bmpsize@grab\bmpsize@temp{1}%
1686   \@bmpsize@skip@one
1687   \ifnum\bmpsize@temp<0\relax
1688     \expandafter\@bmpsize@stop
1689   \fi
1690   \ifnum\bmpsize@temp>1\relax
1691     \expandafter\@bmpsize@stop
1692   \fi
1693   \@bmpsize@skip@two
1694   \@bmpsize@num@two\bmpsize@pixelwidth
1695   \@bmpsize@num@two\bmpsize@pixelheight
1696   \@bmpsize@ok
1697   \@bmpsize@stop
1698   \@nil
1699   \@bmpsize@end
1700 }%
1701 ⟨/base⟩
```

2.3 Package bmpsize

```
1702 ⟨*package⟩
1703 \ProvidesPackage{bmpsize}%
1704   [2006/08/24 v1.0 Extract size and resolution data from bitmap files (HO)]
1705 \RequirePackage{ifpdf}
1706 \ifpdf
1707   \PackageInfo{bmpsize}{Superseded by pdfTeX in PDF mode}%
1708   \expandafter\endinput
1709 \fi
1710 \begingroup\expandafter\expandafter\expandafter\endgroup
1711 \expandafter\ifx\csname pdffiledump\endcsname\relax
1712   \PackageError{bmpsize}{%
1713     You need pdfTeX 1.30.0 or newer%
```

```
1714  }{Package loading is aborted.}%
1715  \expandafter\endinput
1716 \fi
1717
1718 \RequirePackage{graphics}
1719 \RequirePackage{keyval}
1720 \RequirePackage{bmpsize-base}
1721
1722 \begingroup\expandafter\expandafter\expandafter\endgroup
1723 \expandafter\ifx\csname PackageWarning\endcsname\relax
1724   \def\@bmpsize@warning#1#2{%
1725     \begingroup
1726       \newlinechar=10 %
1727       \def\MessageBreak{%
1728         ^^J%
1729         (bmpsize) %
1730         \space\space\space\space
1731         \space\space\space\space
1732         \space\space\space\space
1733         \space\space\space
1734       }%
1735       \immediate\write16{%
1736         Package bmpsize Warning: #2 %
1737         on input line \the\inputlineno.%
1738       }%
1739     \endgroup
1740   }%
1741 \else
1742   \def\@bmpsize@warning{\PackageWarning{bmpsize}}%
1743 \fi
1744
1745 \InputIfFileExists{bmpsize-\Gin@driver}{}{}
1746
1747 \define@key{Gin}{bmpsizefast}[true]{%
1748   \expandafter\ifx\csname if#1\expandafter\endcsname\csname iftrue\endcsname
1749     \@bmpsize@fasttrue
1750   \else
1751     \@bmpsize@fastfalse
1752   \fi
1753 }
1754 \define@key{Gin}{resolutionunit}{%
1755   \def\bmpsize@unit@default{#1}%
1756 }
1757 \begingroup
1758   \def\x#1{\endgroup
1759     \define@key{Gin}{resolution}{%
1760       \@bmpsize@read@resolution\@bmpsize@user@resolutiontrue##1#1#1\@nil
1761     }%
1762     \define@key{Gin}{defaultresolution}{%
1763       \@bmpsize@read@resolution\@bmpsize@user@resolutionfalse##1#1#1\@nil
1764     }%
1765   }%
1766 \x{ }
1767 \def\@bmpsize@read@resolution#1#2 #3 #4\@nil{%
1768   \ifcase 0\ifx\\#2\\1\fi
1769          \ifnum\pdfstrcmp{#2}{\Gin@exclamation}=\z@
1770            \ifx\\#3\\1\fi
1771            \ifnum\pdfstrcmp{#3}{\Gin@exclamation}=\z@
1772              1%
1773            \fi
1774          \fi
1775     \ifcase\pdfstrcmp{#2}{\Gin@exclamation}\relax
```

46

```
1776        \let\bmpsize@pixelx@default\Gin@exclamation
1777      \else
1778        \edef\bmpsize@pixelx@default{#2}%
1779      \fi
1780      \ifcase\pdfstrcmp{#3}{\Gin@exclamation}\relax
1781        \let\bmpsize@pixely@default\Gin@exclamation
1782      \else
1783        \ifx\\#3\\%
1784          \let\bmpsize@pixely@default\bmpsize@pixelx@default
1785        \else
1786          \edef\bmpsize@pixely@default{#3}%
1787        \fi
1788      \fi
1789      #1%
1790    \else
1791      \PackageError{bmpsize}{%
1792        Wrong syntax for key (default)resolution%
1793      }{%
1794        See package documentation for correct syntax.%
1795      }%
1796    \fi
1797 }
1798 \newcommand*{\bmpsizesetup}{\setkeys{Gin}}
1799
1800 \let\@bmpsize@org@setfile\Gin@setfile
1801 \def\Gin@setfile#1#2#3{%
1802    \ifcase\pdfstrcmp{#1}{bmp}\relax
1803      \expandafter\@firstofone
1804    \else
1805      \expandafter\@gobble
1806    \fi
1807    {%
1808      \bmpsize@okfalse
1809      \edef\bmpsize@ext{\ifx\Gin@ext\relax\Gin@eext\else\Gin@ext\fi}%
1810      \edef\bmpsize@file{\Gin@base\bmpsize@ext}%
1811      \edef\@bmpsize@temp{\bmpsize@ext}%
1812      \@ifundefined{bmpsize@read@\@bmpsize@temp}{%
1813        \@ifundefined{bmpsize@map@\@bmpsize@temp}{}{%
1814          \expandafter\let\expandafter\@bmpsize@temp
1815          \csname bmpsize@map@\@bmpsize@temp\endcsname
1816        }%
1817      }{}%
1818      \@ifundefined{bmpsize@read@\@bmpsize@temp}{%
1819      }{%
1820        \csname bmpsize@read@\@bmpsize@temp\endcsname\bmpsize@file
1821      }%
1822      \ifbmpsize@ok
1823      \else
1824        \@for\@bmpsize@temp:=\bmpsize@types\do{%
1825          \ifbmpsize@ok
1826          \else
1827            \csname bmpsize@read@\@bmpsize@temp\endcsname\bmpsize@file
1828          \fi
1829        }%
1830      \fi
1831      \ifbmpsize@ok
1832        \ifGin@bbox
1833          \@ifundefined{Gin@vllx}{%
1834            \@bmpsize@warning{Explicit bounding box is ignored}%
1835          }{%
1836            \ifx\Gin@viewport@code\relax
1837              \def\Gin@ollx{0}%
```

47

```
1838            \let\Gin@olly\Gin@ollx
1839            \let\Gin@ourx\bmpsize@width
1840            \let\Gin@oury\bmpsize@height
1841            \let\Gin@vllx\Gin@llx
1842            \let\Gin@vlly\Gin@lly
1843            \let\Gin@vurx\Gin@urx
1844            \let\Gin@vury\Gin@ury
1845            \let\Gin@viewport@code\Gin@viewport
1846            \@bmpsize@warning{%
1847              Explicit bounding box replaced by\MessageBreak
1848              viewport setting
1849            }%
1850          \else
1851            \@bmpsize@warning{Explicit bounding box is ignored}%
1852          \fi
1853        }%
1854      \fi
1855      \def\Gin@llx{0}%
1856      \def\Gin@lly{0}%
1857      \let\Gin@urx\bmpsize@width
1858      \let\Gin@ury\bmpsize@height
1859      \Gin@bboxtrue
1860    \else
1861      \PackageInfo{bmpsize}{Unknown image type of \bmpsize@file}%
1862    \fi
1863  }%
1864  \@bmpsize@org@setfile{#1}{#2}{#3}%
1865 }
1866 \newcommand*{\bmpsize@ext@type}[1]{%
1867  \@namedef{bmpsize@map@#1}%
1868 }
1869 \bmpsize@ext@type{.jpg}{jpg}
1870 \bmpsize@ext@type{.jpe}{jpg}
1871 \bmpsize@ext@type{.jfif}{jpg}
1872 \bmpsize@ext@type{.jpeg}{jpg}
1873 \bmpsize@ext@type{.tif}{tiff}
1874 \bmpsize@ext@type{.tiff}{tiff}
1875 \bmpsize@ext@type{.pcx}{pcx}
1876 \bmpsize@ext@type{.msp}{msp}
1877 \bmpsize@ext@type{.bmp}{bmp}
1878 \bmpsize@ext@type{.png}{png}
1879 \bmpsize@ext@type{.pnm}{pnm}
1880 \bmpsize@ext@type{.pbm}{pnm}
1881 \bmpsize@ext@type{.pgm}{pnm}
1882 \bmpsize@ext@type{.ppm}{pnm}
1883 \bmpsize@ext@type{.pam}{pam}
1884 \bmpsize@ext@type{.xpm}{xpm}
1885 \bmpsize@ext@type{.gif}{gif}
1886 \bmpsize@ext@type{.tga}{tga}
1887 \bmpsize@ext@type{.sgi}{sgi}
1888 ⟨/package⟩
```

2.4 Drivers

2.4.1 dvips

Identification.

```
1889 ⟨*dvips⟩
1890 \ProvidesFile{bmpsize-dvips.def}%
1891   [2006/08/24 v1.0 Graphics bitmap driver for dvips (HO)]
```

Ensure correct catcodes.

```
1892 \expandafter\edef\csname @bmpsize@driver@catcodes\endcsname{%
1893  \catcode44 \the\catcode44 % ,
```

```
1894    \catcode58 \the\catcode58 % :
1895    \catcode60 \the\catcode60 % <
1896    \catcode61 \the\catcode61 % =
1897    \catcode62 \the\catcode62 % >
1898    \catcode64 \the\catcode64 % @
1899 }
1900 \catcode64 11 %
1901 \@makeother\,
1902 \@makeother\:
1903 \@makeother\<
1904 \@makeother\=
1905 \@makeother\>
```

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

```
1906 \def\Ginclude@bmp#1{%
1907    \message{<#1>}%
1908    \raise\Gin@req@height
1909    \hbox to\Gin@req@width{%
```

Clipping support.

```
1910       \ifGin@clip
1911         \vbox to\z@{%
1912           \special{ps:gsave currentpoint}%
1913           \kern\Gin@req@height
1914           \hbox to\z@{%
1915             \kern\Gin@req@width
1916             \special{ps:%
1917               currentpoint
1918               newpath
1919               3 index 3 index moveto
1920               1 index 3 index lineto
1921               2 copy lineto
1922               exch pop exch pop
1923               lineto
1924               closepath
1925               clip
1926             }%
1927             \hss
1928           }%
1929           \vss
1930         }%
1931       \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.

```
1932       \@ifundefined{Gin@ollx}{%
1933         \dimen@\z@
1934       }{%
1935         \ifx\Gin@scalex\Gin@exclamation
1936           \let\Gin@scalex\Gin@scaley
1937         \fi
1938         \ifx\Gin@scaley\Gin@exclamation
1939           \let\Gin@scaley\Gin@scalex
1940         \fi
1941         \@ifundefined{bmpsize@width}{%
1942           \let\bmpsize@width\Gin@ourx
1943           \let\bmpsize@height\Gin@oury
1944         }{}%
1945         \dimen@=\Gin@llx bp\relax
```

```
1946        \dimen@=\Gin@scalex\dimen@
1947        \kern-\dimen@
1948        \advance\Gin@req@width\dimen@
1949        \dimen@=\bmpsize@width bp\relax
1950        \advance\dimen@ by -\Gin@urx bp\relax
1951        \dimen@=\Gin@scalex\dimen@
1952        \advance\Gin@req@width\dimen@
1953        \dimen@=\Gin@lly bp\relax
1954        \dimen@=\Gin@scaley\dimen@
1955        \advance\Gin@req@height\dimen@
1956        \dimen@=\bmpsize@height bp\relax
1957        \advance\dimen@ by -\Gin@ury bp\relax
1958        \dimen@=\Gin@scaley\dimen@
1959        \advance\Gin@req@height\dimen@
1960      }%
1961      \ifdim\dimen@=\z@
1962      \else
1963        \vbox to\z@\bgroup
1964          \kern-\dimen@
1965      \fi
```
The special for the image.
```
1966      \special{em:graph #1,\the\Gin@req@width,\the\Gin@req@height}%
1967      \ifdim\dimen@=\z@
1968      \else
1969          \vss
1970        \egroup
1971      \fi
1972      \ifGin@clip
1973        \special{ps::grestore}%
1974      \fi
1975      \hss
1976    }%
1977 }

1978 \@bmpsize@driver@catcodes
1979 ⟨/dvips⟩
```

2.4.2   dvipdfm

Identification.
```
1980 ⟨*dvipdfm⟩
1981 \ProvidesFile{bmpsize-dvipdfm.def}%
1982    [2006/08/24 v1.0 Graphics bitmap driver for dvipdfm (HO)]
```
Ensure correct catcodes.
```
1983 \expandafter\edef\csname @bmpsize@driver@catcodes\endcsname{%
1984    \catcode44 \the\catcode44 % ,
1985    \catcode46 \the\catcode46 % .
1986    \catcode58 \the\catcode58 % :
1987    \catcode60 \the\catcode60 % <
1988    \catcode61 \the\catcode61 % =
1989    \catcode62 \the\catcode62 % >
1990    \catcode64 \the\catcode64 % @
1991 }
1992 \catcode64 11 %
1993 \@makeother\,
1994 \@makeother\.
1995 \@makeother\:
1996 \@makeother\<
1997 \@makeother\=
1998 \@makeother\>
```
Counter resource to generate unique names for xform objects.

```
1999 \@ifundefined{@bmpsize@count}{%
2000   \csname newcount\endcsname\@bmpsize@count
2001   \@bmpsize@count=\z@
2002 }{}
```

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

```
2003 \begingroup\expandafter\expandafter\expandafter\endgroup
2004 \expandafter\ifx\csname pdfescapestring\endcsname\relax
2005   \def\@bmpsize@pdfescapestring#1{#1}%
2006 \else
2007   \let\@bmpsize@pdfescapestring\pdfescapestring
2008 \fi
```

The size of reused images of dvipdfm 0.13.2c is 1bp. 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

```
2009 \@ifundefined{bmpsize@dvipdfm@factor}{%
2010   \def\bmpsize@dvipdfm@factor{.99626}%
2011 }{}
```

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

```
2012 \def\Ginclude@bmp#1{%
2013   \message{<#1>}%
```

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

```
2014   \ifGin@clip
2015     \global\advance\@bmpsize@count\@ne
2016     \edef\@bmpsize@clip@name{@CLIP@\the\@bmpsize@count}%
2017     \special{
2018       pdf:bxobj \@bmpsize@clip@name\space
2019       width \the\Gin@req@width\space
2020       height \the\Gin@req@height
2021     }%
2022   \fi
```

Support for viewport/trim.

```
2023   \hbox to \z@{%
2024     \@ifundefined{Gin@ollx}{%
2025       \dimen@\z@
2026     }{%
2027       \ifx\Gin@scalex\Gin@exclamation
2028         \let\Gin@scalex\Gin@scaley
2029       \fi
2030       \ifx\Gin@scaley\Gin@exclamation
2031         \let\Gin@scaley\Gin@scalex
2032       \fi
2033       \@ifundefined{bmpsize@width}{%
2034         \let\bmpsize@width\Gin@ourx
2035         \let\bmpsize@height\Gin@oury
2036       }{}%
2037       \dimen@=\Gin@llx bp\relax
2038       \dimen@=\Gin@scalex\dimen@
2039       \kern-\dimen@
2040       \advance\Gin@req@width\dimen@
2041       \dimen@=\bmpsize@width bp\relax
2042       \advance\dimen@ by -\Gin@urx bp\relax
2043       \dimen@=\Gin@scalex\dimen@
2044       \advance\Gin@req@width\dimen@
```

```
2045        \dimen@=\bmpsize@height bp\relax
2046        \advance\dimen@ by -\Gin@ury bp\relax
2047        \dimen@=\Gin@scaley\dimen@
2048        \advance\Gin@req@height\dimen@
2049        \dimen@=\Gin@lly bp\relax
2050        \dimen@=\Gin@scaley\dimen@
2051        \advance\Gin@req@height\dimen@
2052      }%
2053      \ifdim\dimen@=\z@
2054      \else
2055        \vbox to\z@\bgroup
2056          \kern\dimen@
2057      \fi
```

Reuse support, dvipdfm just remember the image. The requested sizes, clipping, . . . do not matter.

```
2058      \edef\@bmpsize@temp{@IMG@\@bmpsize@pdfescapestring{#1}}%
2059      \@ifundefined{\@bmpsize@temp}{%
2060        \global\advance\@bmpsize@count\@ne
2061        \expandafter\xdef\csname\@bmpsize@temp\endcsname{%
2062          \the\@bmpsize@count
2063        }%
2064        \special{%
2065          pdf:image @IMG\csname\@bmpsize@temp\endcsname\space
2066          width \the\Gin@req@width\space
2067          height \the\Gin@req@height\space
2068          depth 0pt (\@bmpsize@pdfescapestring{#1})%
2069        }%
2070      }{%
2071        \special{%
2072          pdf:bt %
2073          xscale \strip@pt\dimexpr
2074            \bmpsize@dvipdfm@factor\Gin@req@width\relax\space
2075          yscale \strip@pt\dimexpr
2076            \bmpsize@dvipdfm@factor\Gin@req@height\relax
2077        }%
2078        \special{pdf:uxobj @IMG\csname\@bmpsize@temp\endcsname}%
2079        \special{pdf:et}%
2080      }%
2081      \ifdim\dimen@=\z@
2082      \else
2083          \vss
2084        \egroup
2085      \fi
2086      \hss
2087    }%
2088    \ifGin@clip
2089      \special{pdf:exobj}%
2090      \special{pdf:uxobj \@bmpsize@clip@name}%
2091    \fi
2092 }

2093 \@bmpsize@driver@catcodes
2094 ⟨/dvipdfm⟩
```

2.4.3   dvipdfmx

Identification.

```
2095 ⟨*dvipdfmx⟩
2096 \ProvidesFile{bmpsize-dvipdfmx.def}%
2097   [2006/08/24 v1.0 Graphics bitmap driver for dvipdfmx (HO)]
```

Ensure correct catcodes.

```
2098 \expandafter\edef\csname @bmpsize@driver@catcodes\endcsname{%
2099   \catcode46 \the\catcode46 % .
2100   \catcode64 \the\catcode64 % @
2101 }
2102 \catcode64 11 %
2103 \@makeother\.
```

The size of reused images of dvipdfmx-20050823 is 1in in opposite the 1bp of
dvipdfm. The reused image must be scaled to the requested width and height.
The factor is the conversion from in to pt (1/72.27).

\bmpsize@dvipdfm@factor

```
2104 \@ifundefined{bmpsize@dvipdfm@factor}{%
2105   \def\bmpsize@dvipdfm@factor{.01384}%
2106 }{}
```

The rest is the same as for dvipdfm.

```
2107 \@bmpsize@driver@catcodes
2108 \input{bmpsize-dvipdfm.def}
2109 ⟨/dvipdfmx⟩
```

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

```
2110 ⟨∗test⟩
2111 \expandafter\ifx\csname NeedsTeXFormat\endcsname\relax
2112 \input miniltx
2113 \fi
2114 \begingroup\expandafter\expandafter\expandafter\endgroup
2115 \expandafter\ifx\csname pdfoutput\endcsname\relax
2116 \else
2117   \pdfoutput=0 %
2118 \fi
2119 \RequirePackage{bmpsize}
2120
2121 \endlinechar=-1
2122 \catcode`\@=11
2123 \def\msg#{\immediate\write16}
2124
2125 \def\init{%
2126   \msg{}%
2127   \msg{File name menu}%
2128   \msg{==============}%
2129   \msg{* Option menu: use 'opt' as file name}%
2130   \msg{* Quit program: <return>}%
2131   \msg{}%
2132   \message{Image file name = }%
2133   \read-1 to \imagename
2134   \ifx\imagename\@empty
2135     \expandafter\@firstoftwo
2136   \else
2137     \expandafter\@secondoftwo
2138   \fi
2139   {%
2140     \csname @@end\endcsname
2141     \end
2142   }{%
2143     \ifnum\pdfstrcmp{\imagename}{opt}=\z@
2144       \expandafter\optionmenu
2145     \else
2146       \startimg
2147       \expandafter\init
2148     \fi
2149   }%
```

```
2150 }
2151 \def\optionmenu{%
2152   \msg{}%
2153   \msg{Option menu}%
2154   \msg{===========}%
2155   \msg{Current setting:}%
2156   \msg{* bmpsizefast = \if@bmpsize@fast true\else false\fi}%
2157   \msg{* \if@bmpsize@user@resolution\else default\fi resolution = %
2158     \bmpsize@pixelx@default
2159     \space
2160     \bmpsize@pixely@default
2161   }%
2162   \msg{* \if@bmpsize@user@resolution default\fi resolution: not set}%
2163   \msg{* resolutionunit = \bmpsize@unit@default}%
2164   \msg{* Quit option menu: <return>}%
2165   \msg{}%
2166   \message{Options = }%
2167   \read-1 to \options
2168   \ifx\options\empty
2169     \expandafter\init
2170   \else
2171     \edef\@bmpsize@temp{%
2172       \noexpand\setkeys{Gin}{\options}%
2173     }%
2174     \@bmpsize@temp
2175     \expandafter\optionmenu
2176   \fi
2177 }
2178
2179 \def\startimg{%
2180   \let\@found\@empty
2181   \msg{}%
2182   \msg{* File [\imagename]}%
2183   \@for\@type:=\bmpsize@types\do{%
2184     \ifx\@found\@empty
2185       \csname bmpsize@read@\@type\endcsname\imagename
2186       \ifbmpsize@ok
2187         \let\@found\@type
2188         \msg{\space\space Type: \@type}%
2189         \msg{\space\space Pixel width: \bmpsize@pixelwidth\space px}%
2190         \msg{\space\space Pixel height: \bmpsize@pixelheight\space px}%
2191         \ifx\bmpsize@pixelx\relax
2192         \else
2193           \ifx\bmpsize@unit\relax
2194             \let\@unit@spec\@empty
2195             \def\@ratio@name{Ratio }%
2196           \else
2197             \def\@unit@spec{\space dots per \bmpsize@unit}%
2198             \def\@ratio@name{Density }%
2199           \fi
2200           \msg{\space\space \@ratio@name x: \bmpsize@pixelx\@unit@spec}%
2201           \msg{\space\space \@ratio@name y: \bmpsize@pixely\@unit@spec}%
2202         \fi
2203         \msg{\space\space Width: \bmpsize@width\space bp}%
2204         \msg{\space\space Height: \bmpsize@height\space bp}%
2205       \fi
2206     \fi
2207   }%
2208   \ifx\@found\@empty
2209     \edef\@file@date{\pdffilemoddate{\imagename}}%
2210     \ifx\@file@date\@empty
2211       \msg{\space\space --> File not found <--}%
```

```
2212    \else
2213      \msg{\space\space --> Unknown image type <--}%
2214    \fi
2215  \fi
2216 }
2217
2218 \ifx\noinit!\else\expandafter\init\fi
2219 ⟨/test⟩
```

3 Installation

CTAN. 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.

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`.

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

3.1 Some details for the interested

Attached source. The PDF documentation on CTAN also includes the `.dtx` source file. It can be extracted by AcrobatReader 6 or higher. Another option is `pdftk`, e.g. unpack the file into the current directory:

    pdftk bmpsize.pdf unpack_files output .

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.

---

[1] ftp://ftp.ctan.org/tex-archive/

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 References

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

4.1 URLs for bitmap format descriptions

4.1.1 JPEG

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

4.1.2 PNG

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

4.1.3 GIF

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

4.1.4 BMP

- http://en.wikipedia.org/wiki/Windows_bitmap
- http://de.wikipedia.org/wiki/Windows_bitmap
- http://msdn.microsoft.com/library/default.asp?url=/library/en-us/gdi/bitmaps_4v1h.asp
- http://msdn.microsoft.com/library/default.asp?url=/library/en-us/gdi/bitmaps_62uq.asp

4.1.5 PCX

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

4.1.6 MSP

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

4.1.7 TIFF

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

4.1.8 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

4.1.9 SGI

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

4.1.10 WMF

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

4.1.11 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

5 History

[2006/08/24 v1.0]

- First version.

6 Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

Symbols

\, ... 1901, 1993
\. 1994, 2103
\: 1902, 1995
\< 1903, 1996
\= 1904, 1997
\> 1905, 1998
\@ 2122
\@bmpsize@@swap 134, 137
\@bmpsize@abs@byte 171, 180, 187
\@bmpsize@abs@maybe 167, 193, 201, 209
\@bmpsize@absnumfalse 28, 793
\@bmpsize@absnumtrue 791
\@bmpsize@append . . . .
. . . . 82, 1054, 1094, 1211,
1235, 1289, 1438, 1455, 1487, 1504

\@bmpsize@beautify . 220, 222, 391, 392
\@bmpsize@bigendianfalse . . . .
586, 753, 807, 850, 1528, 1598, 1639
\@bmpsize@bigendiantrue . . . .
. . . . 27, 406, 692, 856, 1677
\@bmpsize@break . . . 44, 614, 1025,
1033, 1039, 1062, 1080, 1102,
1158, 1193, 1214, 1228, 1271,
1292, 1306, 1323, 1360, 1397,
1405, 1414, 1439, 1463, 1488, 1512
\@bmpsize@buf 57, 59, 64,
70, 74, 86, 142, 145, 148, 150, 155
\@bmpsize@check@byte . . . 59, 116, 127
\@bmpsize@cleanup@end . 119, 129, 163
\@bmpsize@cleanup@frac 226, 232
\@bmpsize@cleanup@fracdigits 236, 239

59

61