The bmpsize package

Heiko Oberdiek*

2019/12/29 v1.8

Abstract

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

Contents

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

1 Documentation

1.1 Introduction

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

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

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

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

1.2 Bitmap image parsers

This project supports the following image types:

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

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

1.2.1 User interface

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

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

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

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

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

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

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

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

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

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

1.2.2 Hints

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

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

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

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

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

1.2.3 Test program

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

1.2.4 Interface for programmers

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

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

1.3 Improved bitmap inclusion

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

- Support for viewport and trim.
- Support for clip.
- In case of dvipdfm and dvipdfmx the bitmap images are reused and not included again if they are used more than once.

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

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

2 Implementation

2.1 Basic package bmpsize-base

Identification.

```
1 ⟨*base⟩
2 \ProvidesPackage{bmpsize-base}%
3   [2019/12/29 v1.8 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
```

```
  \let\iterate\relax
}
\RequirePackage{pdftexcmds}[2007/11/11]
\newif\ifbmpsize@ok
\let\@bmpsize@ok\bmpsize@oktrue

\newif\if@bmpsize@bigendian
\newif\if@bmpsize@absnum
\newif\if@bmpsize@user@resolution
\newif\if@bmpsize@fast
\@bmpsize@fasttrue

\def\@bmpsize@init{%
  \let\@bmpsize@org@plain@loop\loop
  \let\loop\@bmpsize@plain@loop
  \bmpsize@okfalse
  \@bmpsize@bigendiantrue
  \@bmpsize@absnumfalse
  \let\bmpsize@pixelwidth\relax
  \let\bmpsize@pixelheight\relax
  \let\bmpsize@pixelx\relax
  \let\bmpsize@pixely\relax
  \let\bmpsize@unit\relax
  \let\bmpsize@pixelxdenom\relax
  \let\bmpsize@pixelydenom\relax
  \let\bmpsize@orientation\relax
}

\def\@bmpsize@stop#1\@nil{}

\def\@bmpsize@loop#1{%
  #1%
  \@bmpsize@loop{#1}%
}
\def\@bmpsize@break#1\@bmpsize@loop#2{}

\def\@bmpsize@size#1#2#3{%
  \edef#3{\pdf@filesize{#1}}%
  \ifx#3\@empty
    \expandafter\@bmpsize@stop
  \fi
  \ifnum#3<#2\relax
    \expandafter\@bmpsize@stop
  \fi
}

\def\@bmpsize@read#1#2#3{%
  \edef\@bmpsize@buf{\pdf@filedump{#3}{#2}{#1}}%
  \edef\@bmpsize@temp{%
    \noexpand\@bmpsize@check@byte{#2}\@bmpsize@buf{}{}\noexpand\\%
  }%
  \@bmpsize@temp
}
\def\@bmpsize@fillbuf#1{%
  \ifx\@bmpsize@buf\@empty
    \expandafter\@firstofone
  \else
    \expandafter\@gobble
  \fi
  {%
    \edef\@bmpsize@buf{%
      \pdf@filedump{\bmpsize@offset}{\bmpsize@fillbuflength}{#1}%
```

```
    }%
    \ifx\@bmpsize@buf\@empty
      \expandafter\@bmpsize@stop
    \fi
    \edef\bmpsize@offset{\the\numexpr\bmpsize@offset+\bmpsize@fillbuflength}%
  }%
}
\def\bmpsize@fillbuflength{10}

\def\@bmpsize@append#1#2#3{%
  \edef#1{#2#3}%
}
\def\@bmpsize@pushback#1{%
  \edef\@bmpsize@buf{#1\@bmpsize@buf}%
}

\def\@bmpsize@iswhite#1{%
  \ifnum\pdf@strcmp{#1}{09}=\z@
  \else
    \ifnum\pdf@strcmp{#1}{0A}=\z@
    \else
      \ifnum\pdf@strcmp{#1}{0D}=\z@
      \else
        \ifnum\pdf@strcmp{#1}{20}=\z@
        \else
          1%
        \fi
      \fi
    \fi
  \fi
  \space
}
\def\@bmpsize@isdigit#1{%
  \ifnum\pdf@strcmp{#1}{30}<\z@
    1%
  \else
    \ifnum\pdf@strcmp{#1}{39}>\z@
      1%
    \fi
  \fi
  \space
}

\def\@bmpsize@check@byte#1#2#3{%
  \ifnum#1<\@ne
    \csname fi\endcsname
    \@bmpsize@cleanup@end
  \else
    \csname fi\endcsname
  \ifx!#2#3!%
    \csname fi\endcsname
    \@bmpsize@stop
  \else
    \csname fi\endcsname
    \expandafter\@bmpsize@check@byte\expandafter{\the\numexpr#1-1}%
}
\def\@bmpsize@cleanup@end#1\\{}

\def\@bmpsize@swap@maybe#1{%
  \if@bmpsize@bigendian
  \else
    \edef#1{\expandafter\@bmpsize@@swap#1\@empty\@empty\@empty\@empty}%
```

```
  \fi
}
\def\@bmpsize@@swap#1#2#3#4#5#6#7#8{%
  #7#8#5#6#3#4#1#2%
}

\def\@bmpsize@skip@one{%
  \edef\@bmpsize@buf{\expandafter\@gobbletwo\@bmpsize@buf}%
}
\def\@bmpsize@skip@two{%
  \edef\@bmpsize@buf{\expandafter\@gobblefour\@bmpsize@buf}%
}
\def\@bmpsize@skip@four{%
  \edef\@bmpsize@buf{%
    \expandafter\expandafter\expandafter\@gobblefour\expandafter
    \@gobblefour\@bmpsize@buf
  }%
}

\def\@bmpsize@grab#1#2{%
  \edef#1{\noexpand\@bmpsize@grab@byte#2=\@bmpsize@buf\noexpand\\}%
  \edef#1{#1}%
}
\def\@bmpsize@grab@byte#1=#2#3{%
  #2#3%
  \ifnum#1>\@ne
    \expandafter\@bmpsize@grab@byte\the\numexpr#1-1\expandafter=%
  \else
    \expandafter\@bmpsize@cleanup@end
  \fi
}

\def\@bmpsize@abs@maybe#1{%
  \let\@bmpsize@temp\relax
  \if@bmpsize@absnum
    \ifnum"\expandafter\@car#1\@nil>7 %
      \edef#1{\expandafter\@bmpsize@abs@byte#1\relax}%
      \ifnum\pdf@strcmp{#1}{7FFFFFFF}=\z@
        \let\@bmpsize@temp\@bmpsize@stop
      \else
        \def\@bmpsize@temp{\edef#1{\the\numexpr#1+1}}%
      \fi
    \fi
  \fi
}
\def\@bmpsize@abs@byte#1{%
  \ifx#1\relax
  \else
    \ifcase"0#1 %
      F\or E\or D\or C\or B\or A\or 9\or 8\or
      7\or 6\or 5\or 4\or 3\or 2\or 1\or 0%
    \fi
    \expandafter\@bmpsize@abs@byte
  \fi
}

\def\@bmpsize@num@one#1{%
  \@bmpsize@grab#11%
  \@bmpsize@abs@maybe#1%
  \edef#1{\number"#1}%
  \@bmpsize@temp
  \@bmpsize@skip@one
```

```
}
\def\@bmpsize@num@two#1{%
  \@bmpsize@grab#12%
  \@bmpsize@swap@maybe#1%
  \@bmpsize@abs@maybe#1%
  \edef#1{\number"#1}%
  \@bmpsize@temp
  \@bmpsize@skip@two
}
\def\@bmpsize@num@four#1{%
  \@bmpsize@grab#14%
  \@bmpsize@swap@maybe#1%
  \@bmpsize@abs@maybe#1%
  \ifnum\pdf@strcmp{#1}{7FFFFFFF}>\z@
    \expandafter\@bmpsize@stop
  \fi
  \edef#1{\number"#1}%
  \@bmpsize@temp
  \@bmpsize@skip@four
}

\def\@bmpsize@div#1#2#3{% #1 := #2/#3
  \FPdiv#1{#2}{#3}%
  \@bmpsize@beautify#1%
}
\def\@bmpsize@beautify#1{%
  \FPifint#1%
    \edef#1{\expandafter\@bmpsize@trunc#1.\@nil}%
  \else
    \edef#1{\expandafter\@bmpsize@cleanup@frac#1.\@nil}%
  \fi
}
\def\@bmpsize@trunc#1.#2\@nil{#1}
% #1 isn't an integer, thus we should have at least one
% necessary digit after the dot
\def\@bmpsize@cleanup@frac#1.#2#3.#4\@nil{%
  #1.#2%
  \ifx\\#3\\%
  \else
    \@bmpsize@cleanup@fracdigits#3000000000\@nil
  \fi
}
\def\@bmpsize@cleanup@fracdigits#1#2#3#4#5#6#7#8#9{%
  \ifcase#9 %
    \ifcase#8 %
      \ifcase#7 %
        \ifcase#6 %
          \ifcase#5 %
            \ifcase #4 %
              \ifcase #3 %
                \ifcase #2 %
                  \ifcase #1 %
                  \else
                    #1%
                  \fi
                \else
                  #1#2%
                \fi
              \else
                #1#2#3%
              \fi
            \else
```

```
              #1#2#3#4%
            \fi
          \else
            #1#2#3#4#5%
          \fi
        \else
          #1#2#3#4#5#6%
        \fi
      \else
        #1#2#3#4#5#6#7%
      \fi
    \else
      #1#2#3#4#5#6#7#8%
    \fi
  \else
    #1#2#3#4#5#6#7#8#9%
  \fi
  \@bmpsize@trunc.%
}

\def\@bmpsize@end{%
  \ifbmpsize@ok
    \ifx\bmpsize@pixelwidth\relax
      \bmpsize@okfalse
    \fi
    \ifx\bmpsize@pixelheight\relax
      \bmpsize@okfalse
    \fi
  \fi
  \ifbmpsize@ok
    \ifnum\bmpsize@pixelwidth>\z@
    \else
      \bmpsize@okfalse
    \fi
    \ifnum\bmpsize@pixelheight>\z@
    \else
      \bmpsize@okfalse
    \fi
  \fi
  \ifbmpsize@ok
    \ifcase 0%
      \ifx\bmpsize@pixelx\relax 1 \fi
      \ifx\bmpsize@pixely\relax 1 \fi
      \ifnum\bmpsize@pixelx>\z@\else 1 \fi
      \ifnum\bmpsize@pixely>\z@\else 1 \fi
      \ifx\bmpsize@pixelxdenom\relax
         \ifx\bmpsize@pixelydenom\relax\else 1 \fi
      \else
        \ifnum\bmpsize@pixelxdenom>\z@\else 1 \fi
      \fi
      \ifx\bmpsize@pixelydenom\relax
      \else
        \ifnum\bmpsize@pixelydenom>\z@\else 1 \fi
      \fi
    \else
      \let\bmpsize@pixelx\relax
      \let\bmpsize@pixely\relax
      \let\bmpsize@unit\relax
      \let\bmpsize@pixelxdenom\relax
      \let\bmpsize@pixelydenom\relax
    \fi
    \ifx\bmpsize@pixelxdenom\relax
```

```
  \else
    \@bmpsize@div\bmpsize@pixelx\bmpsize@pixelx\bmpsize@pixelxdenom
    \@bmpsize@div\bmpsize@pixely\bmpsize@pixely\bmpsize@pixelydenom
    \let\bmpsize@pixelxdenom\relax
    \let\bmpsize@pixelydenom\relax
  \fi
  \ifcase 0\ifx\bmpsize@unit\relax 1\fi
           \if@bmpsize@user@resolution 1\fi
           \relax
    \let\bmpsize@calc@unit\bmpsize@unit
    \let\bmpsize@calc@pixelx\bmpsize@pixelx
    \let\bmpsize@calc@pixely\bmpsize@pixely
  \else
    \let\bmpsize@calc@unit\bmpsize@unit@default
    \let\bmpsize@calc@pixelx\bmpsize@pixelx@default
    \let\bmpsize@calc@pixely\bmpsize@pixely@default
    \ifx\bmpsize@calc@pixely\Gin@exclamation
      \ifx\bmpsize@pixelx\relax
        \let\bmpsize@calc@pixely\bmpsize@calc@pixelx
      \else
        \FPdiv\bmpsize@calc@pixely\bmpsize@calc@pixelx\bmpsize@pixelx
        \FPmul\bmpsize@calc@pixely\bmpsize@calc@pixely\bmpsize@pixely
      \fi
    \else
      \ifx\bmpsize@calc@pixelx\Gin@exclamation
        \ifx\bmpsize@pixelx\relax
          \let\bmpsize@calc@pixelx\bmpsize@calc@pixely
        \else
          \FPdiv\bmpsize@calc@pixelx\bmpsize@calc@pixely\bmpsize@pixely
          \FPmul\bmpsize@calc@pixelx\bmpsize@calc@pixelx\bmpsize@pixelx
        \fi
      \fi
    \fi
  \fi
  \FPdiv\bmpsize@width\bmpsize@pixelwidth\bmpsize@calc@pixelx
  \FPdiv\bmpsize@height\bmpsize@pixelheight\bmpsize@calc@pixely
  % calculation of width and height in bp for package graphics
  % 1in = 72bp = 72.27pt, 72/72.27 = 8/8.03, 1pt = 65536sp
  \if@bmpsize@fast
    \edef\bmpsize@width{%
      \strip@pt\dimexpr.99626\dimexpr
      \bmpsize@width\dimexpr\bmpsize@calc@unit
    }%
    \edef\bmpsize@height{%
      \strip@pt\dimexpr.99626\dimexpr
      \bmpsize@height\dimexpr\bmpsize@calc@unit
    }%
  \else
    \edef\@bmpsize@temp{\number\dimexpr\bmpsize@calc@unit}%
    \ifnum\@bmpsize@temp>100000 %
      \FPmul\@bmpsize@temp\@bmpsize@temp{0.00001}%
      \def\@bmpsize@corr{100000}%
    \else
      \let\@bmpsize@corr\relax
    \fi
    \FPmul\bmpsize@width\bmpsize@width\@bmpsize@temp
    \FPmul\bmpsize@height\bmpsize@height\@bmpsize@temp
    \FPmul\bmpsize@width\bmpsize@width{8}%
    \FPmul\bmpsize@height\bmpsize@height{8}%
    \FPdiv\bmpsize@width\bmpsize@width{8.03}%
    \FPdiv\bmpsize@height\bmpsize@height{8.03}%
    \FPdiv\bmpsize@width\bmpsize@width{65536}%
```

```
      \FPdiv\bmpsize@height\bmpsize@height{65536}%
      \ifx\@bmpsize@corr\relax
      \else
        \FPmul\bmpsize@width\bmpsize@width\@bmpsize@corr
        \FPmul\bmpsize@height\bmpsize@height\@bmpsize@corr
      \fi
      \FPround\bmpsize@width\bmpsize@width{5}%
      \FPround\bmpsize@height\bmpsize@height{5}%
      \@bmpsize@beautify\bmpsize@width
      \@bmpsize@beautify\bmpsize@height
    \fi
  \fi
  \let\loop\@bmpsize@org@plain@loop
}
\def\bmpsize@unit@default{72.27pt}% more accurate than 1in
\def\bmpsize@pixelx@default{72}
\let\bmpsize@pixely@default\Gin@exclamation

\def\bmpsize@types{png,jpg,bmp,gif,tiff,pnm,pam,xpm,tga,pcx,msp,sgi}
⟨/base⟩
```

2.2 Bitmap formats

2.2.1 png

```
begin png
big-endian

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

\bmpsize@read@png

```
⟨*base⟩
\def\bmpsize@read@png#1{%
  \@bmpsize@init
```

```
  \@bmpsize@bigendiantrue
  \@bmpsize@read{#1}{24}{0}%
  \@bmpsize@grab\bmpsize@temp{8}%
  \@bmpsize@skip@four
  \@bmpsize@skip@four
  \ifnum\pdf@strcmp{\bmpsize@temp}{89504E470D0A1A0A}=\z@
  \else
    \expandafter\@bmpsize@stop
  \fi
  \@bmpsize@num@four\bmpsize@length
  \@bmpsize@grab\bmpsize@temp{4}%
  \@bmpsize@skip@four
  \ifnum\pdf@strcmp{\bmpsize@temp}{49484452}=\z@
  \else
    \expandafter\@bmpsize@stop
  \fi
  \@bmpsize@num@four\bmpsize@pixelwidth
  \@bmpsize@num@four\bmpsize@pixelheight
  \@bmpsize@ok
  \edef\bmpsize@offset{\the\numexpr20+\bmpsize@length}%
  \@bmpsize@loop{%
    \@bmpsize@read{#1}{8}{\bmpsize@offset}%
    \@bmpsize@num@four\bmpsize@length
    \@bmpsize@grab\bmpsize@temp{4}%
    \@bmpsize@skip@four
    \ifnum\pdf@strcmp{\bmpsize@temp}{49444154}=\z@
      \expandafter\@firstofone
    \else
      \expandafter\@gobble
    \fi
    {%
      \@bmpsize@stop
    }%
    \ifnum\pdf@strcmp{\bmpsize@temp}{70485973}=\z@
      \expandafter\@firstofone
    \else
      \expandafter\@gobble
    \fi
    {%
      \@bmpsize@read{#1}{9}{\numexpr\bmpsize@offset+8\relax}%
      \@bmpsize@num@four\bmpsize@pixelx
      \@bmpsize@num@four\bmpsize@pixely
      \@bmpsize@grab\bmpsize@temp{1}%
      \@bmpsize@skip@one
      \ifnum\bmpsize@temp=1\relax
        \expandafter\@firstofone
      \else
        \expandafter\@gobble
      \fi
      {%
        \def\bmpsize@unit{100cm}%
      }%
      \@bmpsize@stop
    }%
    \edef\bmpsize@offset{\the\numexpr\bmpsize@offset+12+\bmpsize@length}%
  }%
  \@bmpsize@stop
  \@nil
  \@bmpsize@end
}%
⟨/base⟩
```

2.2.2 jpg

```
begin jpg

read 3 0
grab 3      -> $temp % SOI and 0xFF
check streq $temp [0xFF 0xD8 0xFF]
assign {2} -> $offset
assign {0} -> $exifdensity
loop
  read 4 $offset
  grab 1    -> $temp
  check streq $temp [0xFF]
  num 1    -> $temp
  if numeq $temp 0xDA % SOS
    stop
  fi
  % look for JFIF APP0 segment
  if numeq $temp 0xE0 % APP0
    num 2       -> $length
    if numeq $exifdensity 0
      if numge $length 16 % a JFIF segment has 16 bytes at least
        read 12 numexpr($offset + 4)
        grab 5      -> $temp % identifier
        if streq $temp ["JFIF" 0x0]
          check numge $length 16
          skip 2 % version
          num 1       -> $temp % units
          if numeq $temp 1
            assign {72.27pt} -> $unit
          else
            if numeq $temp 2
              assign {1cm} -> $unit
            fi
          fi
          num 2    -> $pixelx
          num 2    -> $pixely
        fi
      fi
    fi
  else
    if numeq $temp 0xE1 % APP1
      % look for Exif APP1 segment
      num 2 -> $length
      if numge $length 20 % identifier (6) + Tiff header (8) + first IFD (>=6)
        read 20 numexpr($offset + 4)
        grab 6 -> $temp
        if streq $temp ["Exif" 0x0 0x0]
          assign numexpr($offset + 10) -> $exifoffset
          % read TIFF header
          grab 2 -> $temp
          if streq $temp ["II"]
            little-endian
          else
            check streq $temp ["MM"]
            % big-endian
          fi
          num 2 -> $temp
          check numeq $temp 42
          num 4 -> $temp % offset of first IFD
          check numgt $temp 0
          % read first IFD
          assign numexpr($temp + $exifoffset) -> $off
```

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

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

\bmpsize@read@jpg

```
⟨*base⟩
\def\bmpsize@read@jpg#1{%
  \@bmpsize@init
```

```
\@bmpsize@read{#1}{3}{0}%
\@bmpsize@grab\bmpsize@temp{3}%
\@bmpsize@skip@two
\@bmpsize@skip@one
\ifnum\pdf@strcmp{\bmpsize@temp}{FFD8FF}=\z@
\else
  \expandafter\@bmpsize@stop
\fi
\def\bmpsize@offset{2}%
\def\bmpsize@exifdensity{0}%
\@bmpsize@loop{%
  \@bmpsize@read{#1}{4}{\bmpsize@offset}%
  \@bmpsize@grab\bmpsize@temp{1}%
  \@bmpsize@skip@one
  \ifnum\pdf@strcmp{\bmpsize@temp}{FF}=\z@
  \else
    \expandafter\@bmpsize@stop
  \fi
  \@bmpsize@num@one\bmpsize@temp
  \ifnum\bmpsize@temp=218\relax
    \expandafter\@firstofone
  \else
    \expandafter\@gobble
  \fi
  {%
    \@bmpsize@stop
  }%
  \ifnum\bmpsize@temp=224\relax
    \expandafter\@firstoftwo
  \else
    \expandafter\@secondoftwo
  \fi
  {%
    \@bmpsize@num@two\bmpsize@length
    \ifnum\bmpsize@exifdensity=0\relax
      \expandafter\@firstofone
    \else
      \expandafter\@gobble
    \fi
    {%
      \unless\ifnum\bmpsize@length<16\relax
        \expandafter\@firstofone
      \else
        \expandafter\@gobble
      \fi
      {%
        \@bmpsize@read{#1}{12}{\numexpr\bmpsize@offset+4\relax}%
        \@bmpsize@grab\bmpsize@temp{5}%
        \@bmpsize@skip@four
        \@bmpsize@skip@one
        \ifnum\pdf@strcmp{\bmpsize@temp}{4A46494600}=\z@
          \expandafter\@firstofone
        \else
          \expandafter\@gobble
        \fi
        {%
          \ifnum\bmpsize@length<16\relax
            \expandafter\@bmpsize@stop
          \fi
          \@bmpsize@skip@two
          \@bmpsize@num@one\bmpsize@temp
          \ifnum\bmpsize@temp=1\relax
```

```
            \expandafter\@firstoftwo
          \else
            \expandafter\@secondoftwo
          \fi
          {%
            \def\bmpsize@unit{72.27pt}%
          }{%
            \ifnum\bmpsize@temp=2\relax
              \expandafter\@firstofone
            \else
              \expandafter\@gobble
            \fi
            {%
              \def\bmpsize@unit{1cm}%
            }%
          }%
          \@bmpsize@num@two\bmpsize@pixelx
          \@bmpsize@num@two\bmpsize@pixely
        }%
      }%
    }%
  }{%
    \ifnum\bmpsize@temp=225\relax
      \expandafter\@firstoftwo
    \else
      \expandafter\@secondoftwo
    \fi
    {%
      \@bmpsize@num@two\bmpsize@length
      \unless\ifnum\bmpsize@length<20\relax
        \expandafter\@firstofone
      \else
        \expandafter\@gobble
      \fi
      {%
        \@bmpsize@read{#1}{20}{\numexpr\bmpsize@offset+4\relax}%
        \@bmpsize@grab\bmpsize@temp{6}%
        \@bmpsize@skip@four
        \@bmpsize@skip@two
        \ifnum\pdf@strcmp{\bmpsize@temp}{457869660000}=\z@
          \expandafter\@firstofone
        \else
          \expandafter\@gobble
        \fi
        {%
          \edef\bmpsize@exifoffset{\the\numexpr\bmpsize@offset+10}%
          \@bmpsize@grab\bmpsize@temp{2}%
          \@bmpsize@skip@two
          \ifnum\pdf@strcmp{\bmpsize@temp}{4949}=\z@
            \expandafter\@firstoftwo
          \else
            \expandafter\@secondoftwo
          \fi
          {%
            \@bmpsize@bigendianfalse
          }{%
            \ifnum\pdf@strcmp{\bmpsize@temp}{4D4D}=\z@
            \else
              \expandafter\@bmpsize@stop
            \fi
          }%
          \@bmpsize@num@two\bmpsize@temp
```

```
595             \ifnum\bmpsize@temp=42\relax
596             \else
597               \expandafter\@bmpsize@stop
598             \fi
599             \@bmpsize@num@four\bmpsize@temp
600             \ifnum\bmpsize@temp>0\relax
601             \else
602               \expandafter\@bmpsize@stop
603             \fi
604             \edef\bmpsize@off{\the\numexpr\bmpsize@temp+\bmpsize@exifoffset}%
605             \@bmpsize@read{#1}{2}{\bmpsize@off}%
606             \@bmpsize@num@two\bmpsize@entries
607             \edef\bmpsize@off{\the\numexpr\bmpsize@off+2}%
608             \@bmpsize@loop{%
609               \ifnum\bmpsize@entries=0\relax
610                 \expandafter\@firstofone
611               \else
612                 \expandafter\@gobble
613               \fi
614               {%
615                 \@bmpsize@break
616               }%
617               \edef\bmpsize@entries{\the\numexpr\bmpsize@entries-1}%
618               \@bmpsize@read{#1}{12}{\bmpsize@off}%
619               \edef\bmpsize@off{\the\numexpr\bmpsize@off+12}%
620               \@bmpsize@num@two\bmpsize@tag
621               \ifnum\bmpsize@tag=296\relax
622                 \expandafter\@firstofone
623               \else
624                 \expandafter\@gobble
625               \fi
626               {%
627                 \@bmpsize@skip@four
628                 \@bmpsize@skip@two
629                 \@bmpsize@num@two\bmpsize@temp
630                 \ifcase\bmpsize@temp\relax
631                 \or
632                   \let\bmpsize@unit\relax
633                 \or
634                   \def\bmpsize@unit{72.27pt}%
635                 \or
636                   \def\bmpsize@unit{1cm}%
637                 \else
638                   \let\bmpsize@unit\relax
639                 \fi
640                 \ifcase\bmpsize@temp\relax
641                 \or
642                 \or
643                   \def\bmpsize@exifdensity{1}%
644                 \or
645                   \def\bmpsize@exifdensity{1}%
646                 \else
647                   \let\bmpsize@exifdensity\bmpsize@exifdensity
648                 \fi
649               }%
650               \ifnum\bmpsize@tag=274\relax
651                 \expandafter\@firstofone
652               \else
653                 \expandafter\@gobble
654               \fi
655               {%
656                 \@bmpsize@skip@four
```

```
657               \@bmpsize@skip@two
658               \@bmpsize@num@two\bmpsize@temp
659               \unless\ifnum\bmpsize@temp<0\relax
660                 \expandafter\@firstofone
661               \else
662                 \expandafter\@gobble
663               \fi
664               {%
665                 \unless\ifnum\bmpsize@temp>8\relax
666                   \expandafter\@firstofone
667                 \else
668                   \expandafter\@gobble
669                 \fi
670                 {%
671                   \let\bmpsize@orientation\bmpsize@temp
672                 }%
673               }%
674             }%
675             \ifnum\bmpsize@tag=282\relax
676               \expandafter\@firstofone
677             \else
678               \expandafter\@gobble
679             \fi
680             {%
681               \@bmpsize@skip@four
682               \@bmpsize@skip@two
683               \@bmpsize@num@four\bmpsize@temp
684               \@bmpsize@read{#1}{8}{\numexpr\bmpsize@temp+\bmpsize@exifoffset\relax}%
685               \@bmpsize@num@four\bmpsize@pixelx
686               \@bmpsize@num@four\bmpsize@temp
687               \ifnum\bmpsize@temp=1\relax
688                 \expandafter\@gobble
689               \else
690                 \expandafter\@firstofone
691               \fi
692               {%
693                 \edef\bmpsize@pixelxdenom{\the\numexpr\bmpsize@temp}%
694               }%
695             }%
696             \ifnum\bmpsize@tag=283\relax
697               \expandafter\@firstofone
698             \else
699               \expandafter\@gobble
700             \fi
701             {%
702               \@bmpsize@skip@four
703               \@bmpsize@skip@two
704               \@bmpsize@num@four\bmpsize@temp
705               \@bmpsize@read{#1}{8}{\numexpr\bmpsize@temp+\bmpsize@exifoffset\relax}%
706               \@bmpsize@num@four\bmpsize@pixely
707               \@bmpsize@num@four\bmpsize@temp
708               \ifnum\bmpsize@temp=1\relax
709                 \expandafter\@gobble
710               \else
711                 \expandafter\@firstofone
712               \fi
713               {%
714                 \edef\bmpsize@pixelydenom{\the\numexpr\bmpsize@temp}%
715               }%
716             }%
717           }%
718           \@bmpsize@bigendiantrue
```

```
      }%
    }%
  }{%
    \edef\bmpsize@temp{\the\numexpr\bmpsize@temp-192}%
    \ifcase\bmpsize@temp\relax
    \or
    \or
    \or
    \or
      \def\bmpsize@temp{-1}%
    \or
    \or
    \or
    \or
      \def\bmpsize@temp{-1}%
    \or
    \or
    \or
    \or
      \def\bmpsize@temp{-1}%
    \or
    \or
    \or
    \else
      \def\bmpsize@temp{-1}%
    \fi
    \ifnum\bmpsize@temp=-1\relax
      \expandafter\@gobble
    \else
      \expandafter\@firstofone
    \fi
    {%
      \@bmpsize@read{#1}{4}{\numexpr\bmpsize@offset+5\relax}%
      \@bmpsize@num@two\bmpsize@pixelheight
      \@bmpsize@num@two\bmpsize@pixelwidth
      \ifnum\bmpsize@pixelheight=0\relax
        \expandafter\@firstofone
      \else
        \expandafter\@gobble
      \fi
      {%
        \let\bmpsize@pixelheight\relax
        \@bmpsize@stop
      }%
      \@bmpsize@ok
      \@bmpsize@stop
    }%
    \@bmpsize@num@two\bmpsize@length
  }%
 }%
 \edef\bmpsize@offset{\the\numexpr\bmpsize@offset+\bmpsize@length+2}%
}%
\@bmpsize@stop
\@nil
\@bmpsize@end
}%
```

⟨/base⟩

2.2.3 bmp

```
begin bmp
little-endian
```

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

\bmpsize@read@bmp

```
⟨*base⟩
\def\bmpsize@read@bmp#1{%
  \@bmpsize@init
  \@bmpsize@bigendianfalse
  \@bmpsize@read{#1}{26}{0}%
  \@bmpsize@grab\bmpsize@temp{2}%
  \@bmpsize@skip@two
  \ifnum\pdf@strcmp{\bmpsize@temp}{424D}=\z@
  \else
    \expandafter\@bmpsize@stop
  \fi
  \@bmpsize@skip@four
  \@bmpsize@skip@four
  \@bmpsize@skip@four
  \@bmpsize@num@two\bmpsize@temp
  \ifnum\bmpsize@temp=12\relax
    \expandafter\@firstofone
  \else
    \expandafter\@gobble
  \fi
  {%
    \@bmpsize@skip@two
    \@bmpsize@num@two\bmpsize@pixelwidth
    \@bmpsize@num@two\bmpsize@pixelheight
    \@bmpsize@ok
    \@bmpsize@stop
  }%
  \ifnum\bmpsize@temp=64\relax
```

```
    \expandafter\@firstofone
  \else
    \expandafter\@gobble
  \fi
  {%
    \@bmpsize@skip@two
    \@bmpsize@num@two\bmpsize@pixelwidth
    \@bmpsize@num@two\bmpsize@pixelheight
    \@bmpsize@ok
    \@bmpsize@stop
  }%
  \@bmpsize@skip@two
  \@bmpsize@num@four\bmpsize@pixelwidth
  \@bmpsize@absnumtrue
  \@bmpsize@num@four\bmpsize@pixelheight
  \@bmpsize@absnumfalse
  \@bmpsize@ok
  \@bmpsize@read{#1}{8}{38}%
  \@bmpsize@num@four\bmpsize@pixelx
  \@bmpsize@num@four\bmpsize@pixely
  \def\bmpsize@unit{100cm}%
  \@bmpsize@stop
  \@nil
  \@bmpsize@end
}%
⟨/base⟩
```

2.2.4 gif

```
begin gif
little-endian

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

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

`\bmpsize@read@gif`

```
⟨*base⟩
\def\bmpsize@read@gif#1{%
  \@bmpsize@init
  \@bmpsize@bigendianfalse
  \@bmpsize@read{#1}{13}{0}%
  \@bmpsize@grab\bmpsize@temp{3}%
  \@bmpsize@skip@two
  \@bmpsize@skip@one
  \ifnum\pdf@strcmp{\bmpsize@temp}{474946}=\z@
  \else
    \expandafter\@bmpsize@stop
```

```
  \fi
  \@bmpsize@skip@two
  \@bmpsize@skip@one
  \@bmpsize@num@two\bmpsize@pixelwidth
  \@bmpsize@num@two\bmpsize@pixelheight
  \@bmpsize@skip@two
  \@bmpsize@num@one\bmpsize@temp
  \ifnum\bmpsize@temp=0\relax
    \expandafter\@gobble
  \else
    \expandafter\@firstofone
  \fi
  {%
    \edef\bmpsize@pixelx{\the\numexpr\bmpsize@temp+15}%
    \def\bmpsize@pixely{64}%
  }%
  \@bmpsize@ok
  \@bmpsize@stop
  \@nil
  \@bmpsize@end
}%
```
⟨/base⟩

2.2.5 tiff

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

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

% First IFD
read 2 $offset
assign numexpr($offset + 2) -> $offset
num 2 -> $entries
ok % must rely on checks at the end
loop
  if numeq $entries 0
    stop
  fi
  assign numexpr($entries - 1) -> $entries
  % entry format:
  % 2 tag
  % 2 field type
  % 4 count
  % 4 value/offset
  read 12 $offset
  assign numexpr($offset + 12) -> $offset
  num 2 -> $tag % tag
  if numeq $temp 296 % ResolutionUnit
    skip 6 % type: 3 (short), count: 1
    num 2 -> $temp
```

```
      ifcase $temp
      or % 1
        clear $unit
      or % 2
        assign {72.27pt} -> $unit
      or % 3
        assign {1cm} -> $unit
      else
        clear $unit
      fi
    fi
    if numeq $tag 256 % ImageWidth
      skip 6
      num 4 -> $pixelwidth
    fi
    if numeq $tag 257 % ImageLength
      skip 6
      num 4 -> $pixelheight
    fi
    if numeq $tag 282 % XResolution
      skip 6
      num 4 -> $temp
      read 8 $temp
      num 4 -> $pixelx
      num 4 -> $temp
      if numeq $temp 1
      else
        assign numexpr($temp) -> $pixelxdenom
        % div $pixelx $temp -> $pixelx
      fi
    fi
    if numeq $tag 283 % YResolution
      skip 6
      num 4 -> $temp
      read 8 $temp
      num 4 -> $pixely
      num 4 -> $temp
      if numeq $temp 1
      else
        assign numexpr($temp) -> $pixelydenom
        % div $pixely $temp -> $pixely
      fi
    fi
  repeat
  end
```

\bmpsize@read@tiff

```
863 ⟨*base⟩
864 \def\bmpsize@read@tiff#1{%
865   \@bmpsize@init
866   \def\bmpsize@unit{72.27pt}%
867   \@bmpsize@read{#1}{8}{0}%
868   \@bmpsize@grab\bmpsize@temp{2}%
869   \@bmpsize@skip@two
870   \ifnum\pdf@strcmp{\bmpsize@temp}{4949}=\z@
871     \expandafter\@firstoftwo
872   \else
873     \expandafter\@secondoftwo
874   \fi
875   {%
876     \@bmpsize@bigendianfalse
877   }{%
878     \ifnum\pdf@strcmp{\bmpsize@temp}{4D4D}=\z@
```

```
    \else
      \expandafter\@bmpsize@stop
    \fi
    \@bmpsize@bigendiantrue
  }%
  \@bmpsize@num@two\bmpsize@temp
  \ifnum\bmpsize@temp=42\relax
  \else
    \expandafter\@bmpsize@stop
  \fi
  \@bmpsize@num@four\bmpsize@offset
  \@bmpsize@read{#1}{2}{\bmpsize@offset}%
  \edef\bmpsize@offset{\the\numexpr\bmpsize@offset+2}%
  \@bmpsize@num@two\bmpsize@entries
  \@bmpsize@ok
  \@bmpsize@loop{%
    \ifnum\bmpsize@entries=0\relax
      \expandafter\@firstofone
    \else
      \expandafter\@gobble
    \fi
    {%
      \@bmpsize@stop
    }%
    \edef\bmpsize@entries{\the\numexpr\bmpsize@entries-1}%
    \@bmpsize@read{#1}{12}{\bmpsize@offset}%
    \edef\bmpsize@offset{\the\numexpr\bmpsize@offset+12}%
    \@bmpsize@num@two\bmpsize@tag
    \ifnum\bmpsize@temp=296\relax
      \expandafter\@firstofone
    \else
      \expandafter\@gobble
    \fi
    {%
      \@bmpsize@skip@four
      \@bmpsize@skip@two
      \@bmpsize@num@two\bmpsize@temp
      \ifcase\bmpsize@temp\relax
      \or
        \let\bmpsize@unit\relax
      \or
        \def\bmpsize@unit{72.27pt}%
      \or
        \def\bmpsize@unit{1cm}%
      \else
        \let\bmpsize@unit\relax
      \fi
    }%
    \ifnum\bmpsize@tag=256\relax
      \expandafter\@firstofone
    \else
      \expandafter\@gobble
    \fi
    {%
      \@bmpsize@skip@four
      \@bmpsize@skip@two
      \@bmpsize@num@four\bmpsize@pixelwidth
    }%
    \ifnum\bmpsize@tag=257\relax
      \expandafter\@firstofone
    \else
      \expandafter\@gobble
```

```
    \fi
    {%
      \@bmpsize@skip@four
      \@bmpsize@skip@two
      \@bmpsize@num@four\bmpsize@pixelheight
    }%
    \ifnum\bmpsize@tag=282\relax
      \expandafter\@firstofone
    \else
      \expandafter\@gobble
    \fi
    {%
      \@bmpsize@skip@four
      \@bmpsize@skip@two
      \@bmpsize@num@four\bmpsize@temp
      \@bmpsize@read{#1}{8}{\bmpsize@temp}%
      \@bmpsize@num@four\bmpsize@pixelx
      \@bmpsize@num@four\bmpsize@temp
      \ifnum\bmpsize@temp=1\relax
        \expandafter\@gobble
      \else
        \expandafter\@firstofone
      \fi
      {%
        \edef\bmpsize@pixelxdenom{\the\numexpr\bmpsize@temp}%
      }%
    }%
    \ifnum\bmpsize@tag=283\relax
      \expandafter\@firstofone
    \else
      \expandafter\@gobble
    \fi
    {%
      \@bmpsize@skip@four
      \@bmpsize@skip@two
      \@bmpsize@num@four\bmpsize@temp
      \@bmpsize@read{#1}{8}{\bmpsize@temp}%
      \@bmpsize@num@four\bmpsize@pixely
      \@bmpsize@num@four\bmpsize@temp
      \ifnum\bmpsize@temp=1\relax
        \expandafter\@gobble
      \else
        \expandafter\@firstofone
      \fi
      {%
        \edef\bmpsize@pixelydenom{\the\numexpr\bmpsize@temp}%
      }%
    }%
  }%
  \@bmpsize@stop
  \@nil
  \@bmpsize@end
}%
⟨/base⟩
```

2.2.6 pnm

```
begin pnm
assign {0} -> $offset
read 3 $offset
assign {3} -> $offset
grab 1 -> $temp
check streq $temp ["P"]
```

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

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

\bmpsize@read@pnm

```
995 ⟨*base⟩
996 \def\bmpsize@read@pnm#1{%
997   \@bmpsize@init
998   \def\bmpsize@offset{0}%
999   \@bmpsize@read{#1}{3}{\bmpsize@offset}%
1000  \def\bmpsize@offset{3}%
1001  \@bmpsize@grab\bmpsize@temp{1}%
1002  \@bmpsize@skip@one
1003  \ifnum\pdf@strcmp{\bmpsize@temp}{50}=\z@
1004  \else
1005    \expandafter\@bmpsize@stop
1006  \fi
1007  \@bmpsize@grab\bmpsize@temp{1}%
1008  \@bmpsize@skip@one
1009  \ifnum\pdf@strcmp{\bmpsize@temp}{31}<\z@
1010    \expandafter\@bmpsize@stop
1011  \fi
1012  \ifnum\pdf@strcmp{\bmpsize@temp}{36}>\z@
1013    \expandafter\@bmpsize@stop
1014  \fi
1015  \@bmpsize@grab\bmpsize@temp{1}%
1016  \@bmpsize@skip@one
1017  \ifcase 0\@bmpsize@iswhite\bmpsize@temp
1018    \expandafter\@gobble
1019  \else
1020    \expandafter\@firstofone
1021  \fi
1022  {%
1023    \@bmpsize@stop
1024  }%
1025  \@bmpsize@loop{%
1026    \@bmpsize@fillbuf{#1}%
1027    \@bmpsize@grab\bmpsize@temp{1}%
1028    \@bmpsize@skip@one
1029    \ifcase 0\@bmpsize@iswhite\bmpsize@temp
1030      \expandafter\@gobble
1031    \else
1032      \expandafter\@firstofone
1033    \fi
1034    {%
1035      \ifnum\pdf@strcmp{\bmpsize@temp}{23}=\z@
1036        \expandafter\@firstoftwo
1037      \else
1038        \expandafter\@secondoftwo
1039      \fi
1040      {%
1041        \@bmpsize@loop{%
1042          \@bmpsize@fillbuf{#1}%
```

```
          \@bmpsize@grab\bmpsize@temp{1}%
          \@bmpsize@skip@one
          \ifnum\pdf@strcmp{\bmpsize@temp}{0A}=\z@
            \expandafter\@firstoftwo
          \else
            \expandafter\@secondoftwo
          \fi
          {%
            \@bmpsize@break
          }{%
            \ifnum\pdf@strcmp{\bmpsize@temp}{0D}=\z@
              \expandafter\@firstofone
            \else
              \expandafter\@gobble
            \fi
            {%
              \@bmpsize@break
            }%
          }%
        }%
      }{%
        \@bmpsize@pushback\bmpsize@temp
        \@bmpsize@break
      }%
    }%
  }%
}%
\def\bmpsize@tempnum{}%
\@bmpsize@loop{%
  \@bmpsize@fillbuf{#1}%
  \@bmpsize@grab\bmpsize@temp{1}%
  \@bmpsize@skip@one
  \ifcase 0\@bmpsize@isdigit\bmpsize@temp
    \expandafter\@firstoftwo
  \else
    \expandafter\@secondoftwo
  \fi
  {%
    \@bmpsize@append\bmpsize@tempnum\bmpsize@tempnum\bmpsize@temp
  }{%
    \ifcase 0\@bmpsize@iswhite\bmpsize@temp
      \expandafter\@firstoftwo
    \else
      \expandafter\@secondoftwo
    \fi
    {%
      \@bmpsize@break
    }{%
      \@bmpsize@stop
    }%
  }%
}%
\edef\bmpsize@pixelwidth{\pdf@unescapehex{\bmpsize@tempnum}}%
\@bmpsize@loop{%
  \@bmpsize@fillbuf{#1}%
  \@bmpsize@grab\bmpsize@temp{1}%
  \@bmpsize@skip@one
  \ifcase 0\@bmpsize@iswhite\bmpsize@temp
    \expandafter\@gobble
  \else
    \expandafter\@firstofone
  \fi
  {%
```

```
      \@bmpsize@pushback\bmpsize@temp
      \@bmpsize@break
    }%
  }%
  \def\bmpsize@tempnum{}%
  \@bmpsize@loop{%
    \@bmpsize@fillbuf{#1}%
    \@bmpsize@grab\bmpsize@temp{1}%
    \@bmpsize@skip@one
    \ifcase 0\@bmpsize@isdigit\bmpsize@temp
      \expandafter\@firstoftwo
    \else
      \expandafter\@secondoftwo
    \fi
    {%
      \@bmpsize@append\bmpsize@tempnum\bmpsize@tempnum\bmpsize@temp
    }{%
      \ifcase 0\@bmpsize@iswhite\bmpsize@temp
        \expandafter\@firstoftwo
      \else
        \expandafter\@secondoftwo
      \fi
      {%
        \@bmpsize@break
      }{%
        \@bmpsize@stop
      }%
    }%
  }%
  \edef\bmpsize@pixelheight{\pdf@unescapehex{\bmpsize@tempnum}}%
  \@bmpsize@ok
  \@bmpsize@stop
  \@nil
  \@bmpsize@end
}%
```
⟨/base⟩

2.2.7 pam

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

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

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

\bmpsize@read@pam

```
1141 ⟨*base⟩
1142 \def\bmpsize@read@pam#1{%
1143   \@bmpsize@init
1144   \@bmpsize@read{#1}{3}{0}%
1145   \def\bmpsize@offset{3}%
1146   \let\bmpsize@off\bmpsize@offset
1147   \@bmpsize@grab\bmpsize@temp{3}%
1148   \@bmpsize@skip@two
1149   \@bmpsize@skip@one
1150   \ifnum\pdf@strcmp{\bmpsize@temp}{50370A}=\z@
1151   \else
```

```
    \expandafter\@bmpsize@stop
  \fi
  \@bmpsize@loop{%
    \@bmpsize@fillbuf{#1}%
    \@bmpsize@grab\bmpsize@temp{1}%
    \@bmpsize@skip@one
    \ifcase 0\@bmpsize@iswhite\bmpsize@temp
      \expandafter\@firstoftwo
    \else
      \expandafter\@secondoftwo
    \fi
    {%
      \edef\bmpsize@off{\the\numexpr\bmpsize@off+1}%
    }{%
      \ifnum\pdf@strcmp{\bmpsize@temp}{23}=\z@
        \expandafter\@firstoftwo
      \else
        \expandafter\@secondoftwo
      \fi
      {%
        \edef\bmpsize@off{\the\numexpr\bmpsize@off+1}%
        \@bmpsize@loop{%
          \@bmpsize@fillbuf{#1}%
          \@bmpsize@grab\bmpsize@temp{1}%
          \@bmpsize@skip@one
          \edef\bmpsize@off{\the\numexpr\bmpsize@off+1}%
          \ifnum\pdf@strcmp{\bmpsize@temp}{0A}=\z@
            \expandafter\@firstofone
          \else
            \expandafter\@gobble
          \fi
          {%
            \@bmpsize@break
          }%
        }%
      }{%
        \@bmpsize@read{#1}{6}{\bmpsize@off}%
        \edef\bmpsize@offset{\the\numexpr\bmpsize@off+6}%
        \@bmpsize@grab\bmpsize@head{5}%
        \@bmpsize@skip@four
        \@bmpsize@skip@one
        \ifnum\pdf@strcmp{\bmpsize@head}{5749445448}=\z@
          \expandafter\@firstoftwo
        \else
          \expandafter\@secondoftwo
        \fi
        {%
          \edef\bmpsize@off{\the\numexpr\bmpsize@off+5}%
          \@bmpsize@loop{%
            \@bmpsize@fillbuf{#1}%
            \@bmpsize@grab\bmpsize@temp{1}%
            \@bmpsize@skip@one
            \ifcase 0\@bmpsize@iswhite\bmpsize@temp
              \expandafter\@firstoftwo
            \else
              \expandafter\@secondoftwo
            \fi
            {%
              \edef\bmpsize@off{\the\numexpr\bmpsize@off+1}%
            }{%
              \ifcase 0\@bmpsize@isdigit\bmpsize@temp
                \expandafter\@firstoftwo
```

```
            \else
              \expandafter\@secondoftwo
            \fi
            {%
              \edef\bmpsize@off{\the\numexpr\bmpsize@off+1}%
              \@bmpsize@break
            }{%
              \@bmpsize@stop
            }%
          }%
        }%
        \let\bmpsize@tempnum\bmpsize@temp
        \@bmpsize@loop{%
          \@bmpsize@fillbuf{#1}%
          \@bmpsize@grab\bmpsize@temp{1}%
          \@bmpsize@skip@one
          \ifcase 0\@bmpsize@isdigit\bmpsize@temp
            \expandafter\@firstoftwo
          \else
            \expandafter\@secondoftwo
          \fi
          {%
            \edef\bmpsize@off{\the\numexpr\bmpsize@off+1}%
            \@bmpsize@append\bmpsize@tempnum\bmpsize@tempnum\bmpsize@temp
          }{%
            \@bmpsize@pushback\bmpsize@temp
            \@bmpsize@break
          }%
        }%
        \@bmpsize@loop{%
          \@bmpsize@fillbuf{#1}%
          \@bmpsize@grab\bmpsize@temp{1}%
          \@bmpsize@skip@one
          \edef\bmpsize@off{\the\numexpr\bmpsize@off+1}%
          \ifnum\pdf@strcmp{\bmpsize@temp}{0A}=\z@
            \expandafter\@firstofone
          \else
            \expandafter\@gobble
          \fi
          {%
            \@bmpsize@break
          }%
        }%
        \edef\bmpsize@pixelwidth{\pdf@unescapehex{\bmpsize@tempnum}}%
      }{%
        \@bmpsize@grab\bmpsize@temp{1}%
        \@bmpsize@skip@one
        \@bmpsize@append\bmpsize@head\bmpsize@head\bmpsize@temp
        \ifnum\pdf@strcmp{\bmpsize@head}{454E44484452}=\z@
          \expandafter\@firstoftwo
        \else
          \expandafter\@secondoftwo
        \fi
        {%
          \@bmpsize@ok
          \@bmpsize@stop
        }{%
          \ifnum\pdf@strcmp{\bmpsize@head}{484549474854}=\z@
            \expandafter\@firstoftwo
          \else
            \expandafter\@secondoftwo
          \fi
```

```
    {%
      \edef\bmpsize@off{\the\numexpr\bmpsize@off+6}%
      \@bmpsize@loop{%
        \@bmpsize@fillbuf{#1}%
        \@bmpsize@grab\bmpsize@temp{1}%
        \@bmpsize@skip@one
        \ifcase 0\@bmpsize@iswhite\bmpsize@temp
          \expandafter\@firstoftwo
        \else
          \expandafter\@secondoftwo
        \fi
        {%
          \edef\bmpsize@off{\the\numexpr\bmpsize@off+1}%
        }{%
          \ifcase 0\@bmpsize@isdigit\bmpsize@temp
            \expandafter\@firstoftwo
          \else
            \expandafter\@secondoftwo
          \fi
          {%
            \edef\bmpsize@off{\the\numexpr\bmpsize@off+1}%
            \@bmpsize@break
          }{%
            \@bmpsize@stop
          }%
        }%
      }%
      \let\bmpsize@tempnum\bmpsize@temp
      \@bmpsize@loop{%
        \@bmpsize@fillbuf{#1}%
        \@bmpsize@grab\bmpsize@temp{1}%
        \@bmpsize@skip@one
        \ifcase 0\@bmpsize@isdigit\bmpsize@temp
          \expandafter\@firstoftwo
        \else
          \expandafter\@secondoftwo
        \fi
        {%
          \edef\bmpsize@off{\the\numexpr\bmpsize@off+1}%
          \@bmpsize@append\bmpsize@tempnum\bmpsize@tempnum\bmpsize@temp
        }{%
          \@bmpsize@pushback\bmpsize@temp
          \@bmpsize@break
        }%
      }%
      \@bmpsize@loop{%
        \@bmpsize@fillbuf{#1}%
        \@bmpsize@grab\bmpsize@temp{1}%
        \@bmpsize@skip@one
        \edef\bmpsize@off{\the\numexpr\bmpsize@off+1}%
        \ifnum\pdf@strcmp{\bmpsize@temp}{0A}=\z@
          \expandafter\@firstofone
        \else
          \expandafter\@gobble
        \fi
        {%
          \@bmpsize@break
        }%
      }%
      \edef\bmpsize@pixelheight{\pdf@unescapehex{\bmpsize@tempnum}}%
    }{%
      \@bmpsize@pushback\bmpsize@head
```

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

2.2.8 xpm

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

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

```
ok
end
```

\bmpsize@read@xpm

```
⟨*base⟩
\def\bmpsize@read@xpm#1{%
  \@bmpsize@init
  \@bmpsize@read{#1}{9}{0}%
  \@bmpsize@grab\bmpsize@temp{9}%
  \@bmpsize@skip@four
  \@bmpsize@skip@four
  \@bmpsize@skip@one
  \def\bmpsize@offset{9}%
  \ifnum\pdf@strcmp{\bmpsize@temp}{2F2A2058504D202A2F}=\z@
  \else
    \expandafter\@bmpsize@stop
  \fi
  \@bmpsize@loop{%
    \@bmpsize@fillbuf{#1}%
    \@bmpsize@grab\bmpsize@temp{1}%
    \@bmpsize@skip@one
    \ifnum\pdf@strcmp{\bmpsize@temp}{22}=\z@
      \expandafter\@firstofone
    \else
      \expandafter\@gobble
    \fi
    {%
      \@bmpsize@break
    }%
    \ifnum\pdf@strcmp{\bmpsize@temp}{2F}=\z@
      \expandafter\@firstofone
    \else
      \expandafter\@gobble
    \fi
    {%
      \@bmpsize@fillbuf{#1}%
      \@bmpsize@grab\bmpsize@temp{1}%
      \@bmpsize@skip@one
      \ifnum\pdf@strcmp{\bmpsize@temp}{2A}=\z@
        \expandafter\@firstofone
      \else
        \expandafter\@gobble
      \fi
      {%
        \@bmpsize@loop{%
          \@bmpsize@fillbuf{#1}%
          \@bmpsize@grab\bmpsize@temp{1}%
          \@bmpsize@skip@one
          \ifnum\pdf@strcmp{\bmpsize@temp}{2A}=\z@
            \expandafter\@firstofone
          \else
            \expandafter\@gobble
          \fi
          {%
            \@bmpsize@loop{%
              \@bmpsize@fillbuf{#1}%
              \@bmpsize@grab\bmpsize@temp{1}%
              \@bmpsize@skip@one
              \ifnum\pdf@strcmp{\bmpsize@temp}{2F}=\z@
                \expandafter\@firstofone
              \else
                \expandafter\@gobble
              \fi
```

```
            {%
              \@bmpsize@break
            }%
            \ifnum\pdf@strcmp{\bmpsize@temp}{2A}=\z@
              \expandafter\@gobble
            \else
              \expandafter\@firstofone
            \fi
            {%
              \@bmpsize@break
            }%
          }%
          \ifnum\pdf@strcmp{\bmpsize@temp}{2F}=\z@
            \expandafter\@firstofone
          \else
            \expandafter\@gobble
          \fi
          {%
            \@bmpsize@break
          }%
        }%
      }%
    }%
  }%
}%
\def\bmpsize@tempnum{}%
\@bmpsize@loop{%
  \@bmpsize@fillbuf{#1}%
  \@bmpsize@grab\bmpsize@temp{1}%
  \@bmpsize@skip@one
  \ifcase 0\@bmpsize@iswhite\bmpsize@temp
    \expandafter\@gobble
  \else
    \expandafter\@firstofone
  \fi
  {%
    \ifcase 0\@bmpsize@isdigit\bmpsize@temp
      \expandafter\@firstoftwo
    \else
      \expandafter\@secondoftwo
    \fi
    {%
      \@bmpsize@append\bmpsize@tempnum\bmpsize@tempnum\bmpsize@temp
      \@bmpsize@break
    }{%
      \@bmpsize@stop
    }%
  }%
}%
\@bmpsize@loop{%
  \@bmpsize@fillbuf{#1}%
  \@bmpsize@grab\bmpsize@temp{1}%
  \@bmpsize@skip@one
  \ifcase 0\@bmpsize@isdigit\bmpsize@temp
    \expandafter\@firstoftwo
  \else
    \expandafter\@secondoftwo
  \fi
  {%
    \@bmpsize@append\bmpsize@tempnum\bmpsize@tempnum\bmpsize@temp
  }{%
    \ifcase 0\@bmpsize@iswhite\bmpsize@temp
```

```
      \expandafter\@firstoftwo
    \else
      \expandafter\@secondoftwo
    \fi
    {%
      \@bmpsize@break
    }{%
      \@bmpsize@stop
    }%
  }%
}%
\edef\bmpsize@pixelwidth{\pdf@unescapehex{\bmpsize@tempnum}}%
\def\bmpsize@tempnum{}%
\@bmpsize@loop{%
  \@bmpsize@fillbuf{#1}%
  \@bmpsize@grab\bmpsize@temp{1}%
  \@bmpsize@skip@one
  \ifcase 0\@bmpsize@iswhite\bmpsize@temp
    \expandafter\@gobble
  \else
    \expandafter\@firstofone
  \fi
  {%
    \ifcase 0\@bmpsize@isdigit\bmpsize@temp
      \expandafter\@firstoftwo
    \else
      \expandafter\@secondoftwo
    \fi
    {%
      \@bmpsize@append\bmpsize@tempnum\bmpsize@tempnum\bmpsize@temp
      \@bmpsize@break
    }{%
      \@bmpsize@stop
    }%
  }%
}%
\@bmpsize@loop{%
  \@bmpsize@fillbuf{#1}%
  \@bmpsize@grab\bmpsize@temp{1}%
  \@bmpsize@skip@one
  \ifcase 0\@bmpsize@isdigit\bmpsize@temp
    \expandafter\@firstoftwo
  \else
    \expandafter\@secondoftwo
  \fi
  {%
    \@bmpsize@append\bmpsize@tempnum\bmpsize@tempnum\bmpsize@temp
  }{%
    \ifcase 0\@bmpsize@iswhite\bmpsize@temp
      \expandafter\@firstoftwo
    \else
      \expandafter\@secondoftwo
    \fi
    {%
      \@bmpsize@break
    }{%
      \@bmpsize@stop
    }%
  }%
}%
\edef\bmpsize@pixelheight{\pdf@unescapehex{\bmpsize@tempnum}}%
\@bmpsize@ok
```

```
  \@bmpsize@stop
  \@nil
  \@bmpsize@end
}%
⟨/base⟩
```

2.2.9 tga

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

\bmpsize@read@tga

```
⟨*base⟩
\def\bmpsize@read@tga#1{%
  \@bmpsize@init
  \@bmpsize@bigendianfalse
  \@bmpsize@read{#1}{16}{1}%
  \@bmpsize@grab\bmpsize@temp{1}%
  \@bmpsize@skip@one
  \ifnum\pdf@strcmp{\bmpsize@temp}{00}=\z@
    \expandafter\@gobble
  \else
    \expandafter\@firstofone
  \fi
  {%
```

```
    \ifnum\pdf@strcmp{\bmpsize@temp}{01}=\z@
      \expandafter\@gobble
    \else
      \expandafter\@firstofone
    \fi
    {%
      \@bmpsize@stop
    }%
  }%
  \@bmpsize@skip@four
  \@bmpsize@skip@four
  \@bmpsize@skip@two
  \@bmpsize@num@two\bmpsize@pixelwidth
  \@bmpsize@num@two\bmpsize@pixelheight
  \@bmpsize@ok
  \@bmpsize@size{#1}{26}\bmpsize@temp  \@bmpsize@read{#1}{26}{\numexpr\bmpsize@temp-26\relax
  \@bmpsize@num@four\bmpsize@offset
  \@bmpsize@skip@four
  \@bmpsize@grab\bmpsize@temp{18}%
  \@bmpsize@skip@four
  \@bmpsize@skip@four
  \@bmpsize@skip@four
  \@bmpsize@skip@four
  \@bmpsize@skip@two
  \ifnum\pdf@strcmp{\bmpsize@temp}{54525545564953494F4E2D5846494C452E00}=\z@
    \expandafter\@gobble
  \else
    \expandafter\@firstofone
  \fi
  {%
    \@bmpsize@stop
  }%
  \ifnum\bmpsize@offset=0\relax
    \expandafter\@firstofone
  \else
    \expandafter\@gobble
  \fi
  {%
    \@bmpsize@stop
  }%
  \@bmpsize@read{#1}{4}{\numexpr\bmpsize@offset+474\relax}%
  \@bmpsize@num@two\bmpsize@pixelx
  \@bmpsize@num@two\bmpsize@pixely
  \ifnum\bmpsize@pixely=0\relax
    \expandafter\@firstofone
  \else
    \expandafter\@gobble
  \fi
  {%
    \let\bmpsize@pixelx\relax
    \let\bmpsize@pixely\relax
  }%
  \@bmpsize@stop
  \@nil
  \@bmpsize@end
}%
⟨/base⟩
```

2.2.10 pcx

```
begin pcx
little-endian
read 16 0
```

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

\bmpsize@read@pcx

```
⟨*base⟩
\def\bmpsize@read@pcx#1{%
  \@bmpsize@init
  \@bmpsize@bigendianfalse
  \@bmpsize@read{#1}{16}{0}%
  \@bmpsize@grab\bmpsize@temp{1}%
  \@bmpsize@skip@one
  \ifnum\pdf@strcmp{\bmpsize@temp}{0A}=\z@
  \else
    \expandafter\@bmpsize@stop
  \fi
  \@bmpsize@skip@one
  \@bmpsize@num@one\bmpsize@temp
  \ifnum\bmpsize@temp=1\relax
  \else
    \expandafter\@bmpsize@stop
  \fi
  \@bmpsize@skip@one
  \@bmpsize@num@two\bmpsize@pixelwidth
  \@bmpsize@num@two\bmpsize@pixelheight
  \@bmpsize@num@two\bmpsize@temp
  \edef\bmpsize@pixelwidth{\the\numexpr\bmpsize@temp-\bmpsize@pixelwidth+1}%
  \@bmpsize@num@two\bmpsize@temp
  \edef\bmpsize@pixelheight{\the\numexpr\bmpsize@temp-\bmpsize@pixelheight+1}%
  \ifnum\bmpsize@pixelwidth>0\relax
  \else
    \expandafter\@bmpsize@stop
  \fi
  \ifnum\bmpsize@pixelheight>0\relax
  \else
    \expandafter\@bmpsize@stop
  \fi
  \@bmpsize@ok
  \@bmpsize@num@two\bmpsize@pixelx
  \@bmpsize@num@two\bmpsize@pixely
  \def\bmpsize@unit{72.27pt}%
  \@bmpsize@stop
  \@nil
  \@bmpsize@end
}%
⟨/base⟩
```

2.2.11 msp

```
begin msp
little-endian

read 16 0

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

\bmpsize@read@msp

```
⟨*base⟩
\def\bmpsize@read@msp#1{%
  \@bmpsize@init
  \@bmpsize@bigendianfalse
  \@bmpsize@read{#1}{16}{0}%
  \@bmpsize@grab\bmpsize@temp{4}%
  \@bmpsize@skip@four
  \ifnum\pdf@strcmp{\bmpsize@temp}{44616E4D}=\z@
    \expandafter\@gobble
  \else
    \expandafter\@firstofone
  \fi
  {%
    \ifnum\pdf@strcmp{\bmpsize@temp}{4C696E53}=\z@
    \else
      \expandafter\@bmpsize@stop
    \fi
  }%
  \@bmpsize@num@two\bmpsize@pixelwidth
  \@bmpsize@num@two\bmpsize@pixelheight
  \@bmpsize@ok
  \@bmpsize@num@two\bmpsize@pixelx
  \@bmpsize@num@two\bmpsize@pixely
  \def\bmpsize@unit{72.27pt}%
  \ifnum\bmpsize@pixelx=0\relax
    \expandafter\@firstofone
  \else
    \expandafter\@gobble
  \fi
  {%
    \@bmpsize@num@two\bmpsize@pixelx
    \@bmpsize@num@two\bmpsize@pixely
  }%
  \@bmpsize@stop
  \@nil
```

```
  \@bmpsize@end
}%
⟨/base⟩
```

2.2.12 sgi

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

\bmpsize@read@sgi

```
⟨*base⟩
\def\bmpsize@read@sgi#1{%
  \@bmpsize@init
  \@bmpsize@bigendiantrue
  \@bmpsize@read{#1}{10}{0}%
  \@bmpsize@grab\bmpsize@temp{2}%
  \@bmpsize@skip@two
  \ifnum\pdf@strcmp{\bmpsize@temp}{01DA}=\z@
  \else
    \expandafter\@bmpsize@stop
  \fi
  \@bmpsize@grab\bmpsize@temp{1}%
  \@bmpsize@skip@one
  \ifnum\bmpsize@temp<0\relax
    \expandafter\@bmpsize@stop
  \fi
  \ifnum\bmpsize@temp>1\relax
    \expandafter\@bmpsize@stop
  \fi
  \@bmpsize@skip@two
  \@bmpsize@num@two\bmpsize@pixelwidth
  \@bmpsize@num@two\bmpsize@pixelheight
  \@bmpsize@ok
  \@bmpsize@stop
  \@nil
  \@bmpsize@end
}%
⟨/base⟩
```

2.3 Package bmpsize

```
⟨*package⟩
\ProvidesPackage{bmpsize}%
  [2019/12/29 v1.8 Extract size/resolution from bitmap files (HO)]%
\RequirePackage{iftex}
\ifpdf
  \PackageInfo{bmpsize}{Superseded by pdfTeX in PDF mode}%
  \expandafter\endinput
\fi
\RequirePackage{pdftexcmds}[2007/11/11]
\begingroup\expandafter\expandafter\expandafter\endgroup
```

```
\expandafter\ifx\csname pdf@filedump\endcsname\relax
  \PackageError{bmpsize}{%
    You need pdfTeX 1.30.0 or newer%
  }{Package loading is aborted.}%
  \expandafter\endinput
\fi

\RequirePackage{infwarerr}[2007/09/09]
\RequirePackage{graphics}
```

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

```
\RequirePackage{keyval}\relax
\expandafter\ifx\csname KV@errx\endcsname\relax
  \def\KV@errx#1{%
    \@PackageError{keyval}{#1}\@ehc
  }%
\fi
\expandafter\ifx\csname KV@err\endcsname\relax
  \let\KV@err\KV@errx
\fi
\RequirePackage{bmpsize-base}

\InputIfFileExists{bmpsize-\Gin@driver}{}{}

\define@key{Gin}{bmpsizefast}[true]{%
  \expandafter\ifx\csname if#1\expandafter\endcsname\csname iftrue\endcsname
    \@bmpsize@fasttrue
  \else
    \@bmpsize@fastfalse
  \fi
}
\define@key{Gin}{resolutionunit}{%
  \def\bmpsize@unit@default{#1}%
}
\begingroup
  \def\x#1{\endgroup
    \define@key{Gin}{resolution}{%
      \@bmpsize@read@resolution\@bmpsize@user@resolutiontrue##1#1#1\@nil
    }%
    \define@key{Gin}{defaultresolution}{%
      \@bmpsize@read@resolution\@bmpsize@user@resolutionfalse##1#1#1\@nil
    }%
  }%
\x{ }
\def\@bmpsize@read@resolution#1#2 #3 #4\@nil{%
  \ifcase 0\ifx\\#2\\1\fi
          \ifnum\pdf@strcmp{#2}{\Gin@exclamation}=\z@
            \ifx\\#3\\1\fi
            \ifnum\pdf@strcmp{#3}{\Gin@exclamation}=\z@
              1%
            \fi
          \fi
    \ifcase\pdf@strcmp{#2}{\Gin@exclamation}\relax
      \let\bmpsize@pixelx@default\Gin@exclamation
    \else
      \edef\bmpsize@pixelx@default{#2}%
    \fi
    \ifcase\pdf@strcmp{#3}{\Gin@exclamation}\relax
      \let\bmpsize@pixely@default\Gin@exclamation
    \else
      \ifx\\#3\\%
        \let\bmpsize@pixely@default\bmpsize@pixelx@default
```

```
      \else
        \edef\bmpsize@pixely@default{#3}%
      \fi
    \fi
    #1%
  \else
    \PackageError{bmpsize}{%
      Wrong syntax for key (default)resolution%
    }{%
      See package documentation for correct syntax.%
    }%
  \fi
}
\newcommand*{\bmpsizesetup}{\setkeys{Gin}}

\let\@bmpsize@org@setfile\Gin@setfile
\def\Gin@setfile#1#2#3{%
  \ifcase\pdf@strcmp{#1}{bmp}\relax
    \expandafter\@firstofone
  \else
    \expandafter\@gobble
  \fi
  {%
    \bmpsize@okfalse
    \edef\bmpsize@ext{\ifx\Gin@ext\relax\Gin@eext\else\Gin@ext\fi}%
    \edef\bmpsize@file{\Gin@base\bmpsize@ext}%
    \edef\@bmpsize@temp{\bmpsize@ext}%
    \@ifundefined{bmpsize@read@\@bmpsize@temp}{%
      \@ifundefined{bmpsize@map@\@bmpsize@temp}{}{%
        \expandafter\let\expandafter\@bmpsize@temp
        \csname bmpsize@map@\@bmpsize@temp\endcsname
      }%
    }{}%
    \@ifundefined{bmpsize@read@\@bmpsize@temp}{%
    }{%
      \csname bmpsize@read@\@bmpsize@temp\endcsname\bmpsize@file
    }%
    \ifbmpsize@ok
    \else
      \@for\@bmpsize@temp:=\bmpsize@types\do{%
        \ifbmpsize@ok
        \else
          \csname bmpsize@read@\@bmpsize@temp\endcsname\bmpsize@file
        \fi
      }%
    \fi
    \ifbmpsize@ok
      \ifGin@bbox
        \@ifundefined{Gin@vllx}{%
          \@PackageWarning{bmpsize}{Explicit bounding box is ignored}%
        }{%
          \ifx\Gin@viewport@code\relax
            \def\Gin@ollx{0}%
            \let\Gin@olly\Gin@ollx
            \let\Gin@ourx\bmpsize@width
            \let\Gin@oury\bmpsize@height
            \let\Gin@vllx\Gin@llx
            \let\Gin@vlly\Gin@lly
            \let\Gin@vurx\Gin@urx
            \let\Gin@vury\Gin@ury
            \let\Gin@viewport@code\Gin@viewport
            \@PackageWarning{bmpsize}{%
```

```
              Explicit bounding box replaced by\MessageBreak
              viewport setting%
            }%
          \else
            \@PackageWarning{bmpsize}{Explicit bounding box is ignored}%
          \fi
        }%
      \fi
      \def\Gin@llx{0}%
      \def\Gin@lly{0}%
      \let\Gin@urx\bmpsize@width
      \let\Gin@ury\bmpsize@height
      \Gin@bboxtrue
    \else
      \PackageInfo{bmpsize}{Unknown image type of \bmpsize@file}%
    \fi
  }%
  \@bmpsize@org@setfile{#1}{#2}{#3}%
}
\newcommand*{\bmpsize@ext@type}[1]{%
  \@namedef{bmpsize@map@#1}%
}
\bmpsize@ext@type{.jpg}{jpg}
\bmpsize@ext@type{.jpe}{jpg}
\bmpsize@ext@type{.jfif}{jpg}
\bmpsize@ext@type{.jpeg}{jpg}
\bmpsize@ext@type{.tif}{tiff}
\bmpsize@ext@type{.tiff}{tiff}
\bmpsize@ext@type{.pcx}{pcx}
\bmpsize@ext@type{.msp}{msp}
\bmpsize@ext@type{.bmp}{bmp}
\bmpsize@ext@type{.png}{png}
\bmpsize@ext@type{.pnm}{pnm}
\bmpsize@ext@type{.pbm}{pnm}
\bmpsize@ext@type{.pgm}{pnm}
\bmpsize@ext@type{.ppm}{pnm}
\bmpsize@ext@type{.pam}{pam}
\bmpsize@ext@type{.xpm}{xpm}
\bmpsize@ext@type{.gif}{gif}
\bmpsize@ext@type{.tga}{tga}
\bmpsize@ext@type{.sgi}{sgi}
⟨/package⟩
```

2.4 Drivers

2.4.1 dvips

Identification.

```
⟨*dvips⟩
\ProvidesFile{bmpsize-dvips.def}%
  [2019/12/29 v1.8 Graphics bitmap driver for dvips (HO)]%
```

Ensure correct catcodes.

```
\expandafter\edef\csname @bmpsize@driver@catcodes\endcsname{%
  \catcode44 \the\catcode44 % ,
  \catcode58 \the\catcode58 % :
  \catcode60 \the\catcode60 % <
  \catcode61 \the\catcode61 % =
  \catcode62 \the\catcode62 % >
  \catcode64 \the\catcode64 % @
}
\catcode64 11 %
\@makeother\,
\@makeother\:
```

```
\@makeother\<
\@makeother\=
\@makeother\>
```

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

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

Clipping support.

```
    \ifGin@clip
      \vbox to\z@{%
        \special{ps:gsave currentpoint}%
        \kern\Gin@req@height
        \hbox to\z@{%
          \kern\Gin@req@width
          \special{ps:%
            currentpoint %
            newpath %
            3 index 3 index moveto %
            1 index 3 index lineto %
            2 copy lineto %
            exch pop exch pop %
            lineto %
            closepath %
            clip %
          }%
          \hss
        }%
        \vss
      }%
    \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.

```
    \@ifundefined{Gin@ollx}{%
      \dimen@\z@
    }{%
      \ifx\Gin@scalex\Gin@exclamation
        \let\Gin@scalex\Gin@scaley
      \fi
      \ifx\Gin@scaley\Gin@exclamation
        \let\Gin@scaley\Gin@scalex
      \fi
      \@ifundefined{bmpsize@width}{%
        \let\bmpsize@width\Gin@ourx
        \let\bmpsize@height\Gin@oury
      }{}%
      \dimen@=\Gin@llx bp\relax
      \dimen@=\Gin@scalex\dimen@
      \kern-\dimen@
      \advance\Gin@req@width\dimen@
      \dimen@=\bmpsize@width bp\relax
      \advance\dimen@ by -\Gin@urx bp\relax
      \dimen@=\Gin@scalex\dimen@
      \advance\Gin@req@width\dimen@
      \dimen@=\Gin@lly bp\relax
      \dimen@=\Gin@scaley\dimen@
```

```
      \advance\Gin@req@height\dimen@
      \dimen@=\bmpsize@height bp\relax
      \advance\dimen@ by -\Gin@ury bp\relax
      \dimen@=\Gin@scaley\dimen@
      \advance\Gin@req@height\dimen@
    }%
    \ifdim\dimen@=\z@
    \else
      \vbox to\z@\bgroup
        \kern-\dimen@
    \fi
```

The special for the image.

```
    \special{em:graph #1,\the\Gin@req@width,\the\Gin@req@height}%
    \ifdim\dimen@=\z@
    \else
        \vss
      \egroup
    \fi
    \ifGin@clip
      \special{ps::grestore}%
    \fi
    \hss
  }%
}

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

2.4.2 dvipdfm and dvipdfmx

Identification.

```
⟨*dvipdfm⟩
\ProvidesFile{bmpsize-dvipdfm.def}%
  [2019/12/29 v1.8 Graphics bitmap driver for dvipdfm (HO)]%
⟨/dvipdfm⟩
⟨*dvipdfmx⟩
\ProvidesFile{bmpsize-dvipdfmx.def}%
  [2019/12/29 v1.8 Graphics bitmap driver for dvipdfmx (HO)]%
⟨/dvipdfmx⟩
⟨*dvipdfm | dvipdfmx⟩
```

Ensure correct catcodes.

```
\expandafter\edef\csname @bmpsize@driver@catcodes\endcsname{%
  \catcode44 \the\catcode44 % ,
  \catcode46 \the\catcode46 % .
  \catcode58 \the\catcode58 % :
  \catcode60 \the\catcode60 % <
  \catcode61 \the\catcode61 % =
  \catcode62 \the\catcode62 % >
  \catcode64 \the\catcode64 % @
}
\catcode64 11 %
\@makeother\,
\@makeother\.
\@makeother\:
\@makeother\<
\@makeother\=
\@makeother\>
```

Counter resource to generate unique names for xform objects.

```
\@ifundefined{@bmpsize@count}{%
  \csname newcount\endcsname\@bmpsize@count
  \@bmpsize@count=\z@
```

```
}{}
```

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

`\@bmpsize@pdfescapestring`

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname pdf@escapestring\endcsname\relax
  \def\@bmpsize@pdfescapestring#1{#1}%
\else
  \let\@bmpsize@pdfescapestring\pdf@escapestring
\fi
```

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

`\bmpsize@dvipdfm@factor`

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

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

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

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

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

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

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

```
  \ifGin@clip
    \global\advance\@bmpsize@count\@ne
    \edef\@bmpsize@clip@name{@CLIP@\the\@bmpsize@count}%
    \special{%
      pdf:bxobj \@bmpsize@clip@name\space
      width \the\Gin@req@width\space
      height \the\Gin@req@height
    }%
  \fi
```

Support for viewport/trim.

```
  \hbox to \z@{%
    \@ifundefined{Gin@ollx}{%
      \dimen@\z@
    }{%
      \ifx\Gin@scalex\Gin@exclamation
        \let\Gin@scalex\Gin@scaley
      \fi
      \ifx\Gin@scaley\Gin@exclamation
        \let\Gin@scaley\Gin@scalex
      \fi
      \@ifundefined{bmpsize@width}{%
        \let\bmpsize@width\Gin@ourx
        \let\bmpsize@height\Gin@oury
      }{}%
      \dimen@=\Gin@llx bp\relax
      \dimen@=\Gin@scalex\dimen@
```

```
      \kern-\dimen@
      \advance\Gin@req@width\dimen@
      \dimen@=\bmpsize@width bp\relax
      \advance\dimen@ by -\Gin@urx bp\relax
      \dimen@=\Gin@scalex\dimen@
      \advance\Gin@req@width\dimen@
      \dimen@=\bmpsize@height bp\relax
      \advance\dimen@ by -\Gin@ury bp\relax
      \dimen@=\Gin@scaley\dimen@
      \advance\Gin@req@height\dimen@
      \dimen@=\Gin@lly bp\relax
      \dimen@=\Gin@scaley\dimen@
      \advance\Gin@req@height\dimen@
    }%
    \ifdim\dimen@=\z@
    \else
      \vbox to\z@\bgroup
        \kern\dimen@
    \fi
```

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

```
    \edef\@bmpsize@temp{@IMG@\@bmpsize@pdfescapestring{#1}}%
    \@ifundefined{\@bmpsize@temp}{%
      \global\advance\@bmpsize@count\@ne
⟨*dvipdfm⟩
      \expandafter\xdef\csname\@bmpsize@temp\endcsname{%
        \the\@bmpsize@count
      }%
⟨/dvipdfm⟩
⟨*dvipdfmx⟩
      \expandafter\ifx\csname bmpsize@pixelwidth\endcsname\relax
      \else
        \expandafter\xdef\csname\@bmpsize@temp\endcsname{%
          \the\@bmpsize@count:\bmpsize@width:\bmpsize@height
        }%
      \fi
⟨/dvipdfmx⟩
      \special{%
        pdf:image @IMG\the\@bmpsize@count\space
        width \the\Gin@req@width\space
        height \the\Gin@req@height\space
        depth 0pt (\@bmpsize@pdfescapestring{#1})%
      }%
    }{%
⟨*dvipdfm⟩
      \special{%
        pdf:bt %
        xscale \strip@pt\dimexpr
          \bmpsize@dvipdfm@factor\Gin@req@width\relax\space
        yscale \strip@pt\dimexpr
          \bmpsize@dvipdfm@factor\Gin@req@height\relax
      }%
      \special{pdf:uxobj @IMG\csname\@bmpsize@temp\endcsname}%
      \special{pdf:et}%
⟨/dvipdfm⟩
⟨*dvipdfmx⟩
      \expandafter\expandafter\expandafter\@bmpsize@extract
          \csname\@bmpsize@temp\endcsname\@nil
      \edef\@bmpsize@xscale{\strip@pt\Gin@req@width}%
      \edef\@bmpsize@temp{\strip@pt\dimexpr\@bmpsize@width bp}%
      \@bmpsize@div\@bmpsize@xscale\@bmpsize@xscale\@bmpsize@temp
      \edef\@bmpsize@yscale{\strip@pt\Gin@req@height}%
```

```
      \edef\@bmpsize@temp{\strip@pt\dimexpr\@bmpsize@height bp}%
      \@bmpsize@div\@bmpsize@yscale\@bmpsize@yscale\@bmpsize@temp
      \special{%
        pdf:bt %
        xscale \@bmpsize@xscale\space
        yscale \@bmpsize@yscale
      }%
      \special{pdf:uxobj @IMG\@bmpsize@imgnum}%
      \special{pdf:et}%
⟨/dvipdfmx⟩
    }%
    \ifdim\dimen@=\z@
    \else
        \vss
      \egroup
    \fi
    \hss
  }%
  \ifGin@clip
    \special{pdf:exobj}%
    \special{pdf:uxobj \@bmpsize@clip@name}%
  \fi
}
⟨*dvipdfmx⟩
\def\@bmpsize@extract#1:#2:#3\@nil{%
  \def\@bmpsize@imgnum{#1}%
  \def\@bmpsize@width{#2}%
  \def\@bmpsize@height{#3}%
}
⟨/dvipdfmx⟩

\@bmpsize@driver@catcodes
⟨/dvipdfm | dvipdfmx⟩
```

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

```
⟨*test⟩
\expandafter\ifx\csname NeedsTeXFormat\endcsname\relax
  \input miniltx\relax
\fi
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname pdfoutput\endcsname\relax
\else
  \pdfoutput=0 %
\fi
\RequirePackage{bmpsize}

\endlinechar=-1
\catcode`\@=11
\def\msg#{\immediate\write16}

\def\init{%
  \msg{}%
  \msg{File name menu}%
  \msg{==============}%
  \msg{* Option menu: use `opt' as file name}%
  \msg{* Quit program: <return>}%
  \msg{}%
  \message{Image file name = }%
  \read-1 to \imagename
  \ifx\imagename\@empty
    \expandafter\@firstoftwo
```

```
  \else
    \expandafter\@secondoftwo
  \fi
  {%
    \csname @@end\endcsname
    \end
  }{%
    \ifnum\pdf@strcmp{\imagename}{opt}=\z@
      \expandafter\optionmenu
    \else
      \startimg
      \expandafter\init
    \fi
  }%
}
\def\optionmenu{%
  \msg{}%
  \msg{Option menu}%
  \msg{===========}%
  \msg{Current setting:}%
  \msg{* bmpsizefast = \if@bmpsize@fast true\else false\fi}%
  \msg{* \if@bmpsize@user@resolution\else default\fi resolution = %
    \bmpsize@pixelx@default
    \space
    \bmpsize@pixely@default
  }%
  \msg{* \if@bmpsize@user@resolution default\fi resolution: not set}%
  \msg{* resolutionunit = \bmpsize@unit@default}%
  \msg{* Quit option menu: <return>}%
  \msg{}%
  \message{Options = }%
  \read-1 to \options
  \ifx\options\empty
    \expandafter\init
  \else
    \edef\@bmpsize@temp{%
      \noexpand\setkeys{Gin}{\options}%
    }%
    \@bmpsize@temp
    \expandafter\optionmenu
  \fi
}

\def\startimg{%
  \let\@found\@empty
  \msg{}%
  \msg{* File [\imagename]}%
  \@for\@type:=\bmpsize@types\do{%
    \ifx\@found\@empty
      \csname bmpsize@read@\@type\endcsname\imagename
      \ifbmpsize@ok
        \let\@found\@type
        \msg{\space\space Type: \@type}%
        \msg{\space\space Pixel width: \bmpsize@pixelwidth\space px}%
        \msg{\space\space Pixel height: \bmpsize@pixelheight\space px}%
        \ifx\bmpsize@pixelx\relax
        \else
          \ifx\bmpsize@unit\relax
            \let\@unit@spec\@empty
            \def\@ratio@name{Ratio }%
          \else
            \def\@unit@spec{\space dots per \bmpsize@unit}%
```

```
            \def\@ratio@name{Density }%
          \fi
          \msg{\space\space \@ratio@name x: \bmpsize@pixelx\@unit@spec}%
          \msg{\space\space \@ratio@name y: \bmpsize@pixely\@unit@spec}%
        \fi
        \msg{\space\space Width: \bmpsize@width\space bp}%
        \msg{\space\space Height: \bmpsize@height\space bp}%
        \ifx\bmpsize@orientation\relax
        \else
          \msg{\space\space Orientation: \bmpsize@orientation}%
        \fi
      \fi
    \fi
  }%
  \ifx\@found\@empty
    \edef\@file@date{\pdf@filemoddate{\imagename}}%
    \ifx\@file@date\@empty
      \msg{\space\space --> File not found <--}%
    \else
      \msg{\space\space --> Unknown image type <--}%
    \fi
  \fi
}

\ifx\noinit!\else\expandafter\init\fi
```
⟨/test⟩

3 Installation

3.1 Download

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

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

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

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

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

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

3.2 Bundle installation

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

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

3.3 Package installation

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

```
tex bmpsize.dtx
```

[1] CTAN:pkg/bmpsize

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

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

3.4 Refresh file name databases

If your TEX distribution (TEX Live, MiKTEX, ...) relies on file name databases, you must refresh these. For example, TEX Live users run `texhash` or `mktexlsr`.

3.5 Some details for the interested

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

plain TEX: Run docstrip and extract the files.

LATEX: Generate the documentation.

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

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

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

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

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

An example follows how to generate the documentation with pdfLATEX:

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

4 References

[1] D. P. Carlisle, The LATEX Project: *Packages in the 'graphics' bundle*, 2005/11/14; CTAN:pkg/grfguide.

4.1 URLs for bitmap format descriptions

4.1.1 JPEG

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

4.1.2 PNG

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

4.1.3 GIF

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

4.1.4 BMP

- https://en.wikipedia.org/wiki/Windows_bitmap
- https://docs.microsoft.com/en-us/windows/win32/gdi/bitmap-storage
- https://docs.microsoft.com/en-us/windows/win32/api/wingdi/ns-wingdi-bitmapfileheader

4.1.5 PCX

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

4.1.6 MSP

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

4.1.7 TIFF

- https://en.wikipedia.org/wiki/TIFF
- https://www.adobe.io/content/dam/udp/en/open/standards/tiff/TIFF6.pdf

4.1.8 TGA

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

4.1.9 SGI

- https://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

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

5 History

[2006/08/24 v1.0]

- First version.

[2007/02/18 v1.1]

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

[2007/04/11 v1.2]

- Line ends sanitized.

[2007/05/01 v1.3]

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

[2007/11/11 v1.4]

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

[2008/08/11 v1.5]

- Code is not changed.
- Update of URLs.

[2009/09/04 v1.6]

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

[2016/05/16 v1.7]

- Documentation updates.

[2019/12/29 v1.8]

- Use iftex package not ifpdf.

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

Symbols

A

B