

Adobe Photoshop File Format Summary




1&url=https%3A%2F%2Fwww.fileformat.info%2Fformat%2Fpsd%2Fegff.htm&text=Adobe%20Photoshop%3A%20Summary%20from%20the

1&url=https%3A%2F%2Fwww.fileformat.info%2Fformat%2Fpsd%2Fegff.htm&text=Adobe%20Photoshop%3A%20Summary%20from%20the

1&url=https%3A%2F%2Fwww.fileformat.info%2Fformat%2Fpsd%2Fegff.htm&text=Adobe%20Photoshop%3A%20Summary%20from%20the

1&url=https%3A%2F%2Fwww.fileformat.info%2Fformat%2Fpsd%2Fegff.htm&text=Adobe%20Photoshop%3A%20Summary%20from%20the

1&url=https%3A%2F%2Fwww.fileformat.info%2Fformat%2Fpsd%2Fegff.htm&text=Adobe%20Photoshop%3A%20Summary%20from%20the



Also Known As: PSD, Adobe PSD, Photoshop 3.0

Type	Bitmap
Colors	Unlimited
Compression	Uncompressed, RLE
Maximum Image Size	30,000x30,000
Multiple Images Per File	No
Numerical Format	Big-endian
Originator	Adobe
Platform	Microsoft Windows, Apple Macintosh
Supporting Applications	Adobe Photoshop, Adobe Premiere, desktop publishing programs
See Also	Macintosh Paint (.../macpaint/index.htm), TIFF (.../tiff/index.htm)

Usage

Storage of images altered or manipulated in the Adobe Photoshop environment.

Comments

A flexible format that is easily read and written but lacks a superior compression scheme. It provides specially good support for different color storage schemes.

Vendor specifications (spec/index.htm) are available for this format.

Sample images (sample/index.htm) are available for this format.

Adobe's Photoshop is probably the fullest featured and most highly respected commercial image-processing bitmap manipulation program in the PC and Macintosh worlds. Its wide distribution has meant that image data is often left in PSD format files and may persist in this form after the original image data is long gone.

**Contents:**  
File Organization  
File Details  
For Further Information

Photoshop users are often professionals working with images in truecolor. Images can be very large, and Photoshop users are expected to have adequate memory to load and manipulate large files. Adobe has chosen to optimize the speed of reading and writing in the Photoshop system, and this is reflected in the RLE compression incorporated in the PSD format. Files, therefore, are not as small as they might be using another compression method. Although this makes sense in the context of a Photoshop-equipped workstation, the lack of a superior compression scheme has probably prevented PSD from becoming more popular as a general-purpose interchange format. Because PSD is an application-specific format, expect it to change in the future.

Under Microsoft Windows, Photoshop files are stored with the PSD suffix and can be identified by looking for the file ID value 8BPS. On the Macintosh, Photoshop files are resource-fork only, and data is recognized by the file ID value 8BPS.

Earlier versions of the format had no compression and were tied to the Macintosh platform.

## File Organization

PSD files consist of a header and three informational sections, called the Mode Block, Image Resources Block, and Layer and Mask Information Block, respectively. These are followed by the actual image data. The header is a series of fixed fields, and the other blocks are variable-length.

Header
Color Mode Data Block
Image Resources Block
Layer and Mask Information Block
Image Data

# File Details

This section describes the details of the various sections of a PSD file.

## Header

The header is 26 bytes in length and is structured as follows:

```
typedef struct _PSD_HEADER
{
    BYTE Signature[4];    /* File ID "8BPS" */
    WORD Version;         /* Version number, always 1 */
    BYTE Reserved[6];     /* Reserved, must be zeroed */
    WORD Channels;         /* Number of color channels (1-24) including alpha
                           channels */

    LONG Rows;            /* Height of image in pixels (1-30000) */
    LONG Columns;         /* Width of image in pixels (1-30000) */
    WORD Depth;           /* Number of bits per channel (1, 8, and 16) */
    WORD Mode;            /* Color mode */
} PSD_HEADER;
```

Only Channels values of 1-16 are supported in v2.5 files. Also in v2.5, only Depth values of 1 and 8 bits per channel are supported.

Data in the mode field specifies how the image is to be interpreted.

### Mode Description

- 0     Bitmap (monochrome)
- 1     Gray-scale
- 2     Indexed color (palette color)
- 3     RGB color
- 4     CMYK color
- 7     Multichannel color
- 8     Duotone (halftone)
- 9     Lab color

## Color Mode Data Block

Following the header is the Color Mode Data block. At the start of this block is a LONG number, in big-endian format, specifying the length of the block in bytes. The nature of the following data is related to the value of the mode field of the header. If the mode field value is anything other than 2 or 6, the length of the block will be zero, and no data will follow the 4-byte length field. If the mode field value is 2, however, signifying indexed color, the following 768 bytes will contain a 256-color palette. If the mode field value is 6, signifying duotone data, the data following presumably consists of screen parameters and other related information. Unfortunately, it is intentionally not documented by Adobe, and non-Photoshop readers are advised to treat duotone images as gray-scale images.

## Image Resources Block

Following the Color Mode Data block is the Image Resources block. Like the Color Mode Data block, the first four bytes are a LONG number in big-endian format specifying the length of the block. The following data consists of non-pixel data associated with an image. Information in the Image Resources block was stored in the resource fork in early Photoshop versions running on the Macintosh platform.

```
struct _ColorModeDataBlock
{
    BYTE Type[4];    /* Always "8BIM" */
    WORD ID;         /* (See table below) */
    BYTE Name[];     /* Even-length Pascal-format string, 2 bytes or longer */
    LONG Size;       /* Length of resource data following, in bytes */
    BYTE Data[];     /* Resource data, padded to even length */
};
```

The format of the data is determined by the value in the ID field, which can have the following values:

ID	Data Format Description	
03e8	WORD[5]	Channels, rows, columns, depth, and mode
03e9		Optional Macintosh print manager information
03eb		Indexed color table
03ed	(See below)	Resolution information
03ee	BYTE[]	Alpha channel names (Pascal-format strings)
03ef	(See below)	Display information for each channel
03f0	BYTE[]	Optional Pascal-format caption string
03f1	LONG, WORD	Fixed-point border width, border units (see below)

03f2		Background color
03f3	BYTE[8]	Print flags (see below)
03f4		Gray-scale and halftoning information
03f5		Color halftoning information
03f6		Duotone halftoning information
03f7		Gray-scale and multichannel transfer function
03f8		Color transfer functions
03f9		Duotone transfer functions
03fa		Duotone image information
03fb	BYTE[2]	Effective black and white value for dot range
03fc		
03fd		EPS options
03fe	WORD, BYTE	Quick Mask channel ID, flag for mask initially empty
03ff		
0400	WORD	Index of target layer (0=bottom)
0401		Working path
0402	WORD[]	Layers group info, group ID for dragging groups
0403		
0404		IPTC-NAA record
0405		Image mode for raw-format files
0406		JPEG quality (Adobe internal)
07d0-0bb6		Saved path information
0bb7		Clipping pathname
2710	(See below)	Print flags information

ID values 03e8, 03eb, 03ff, and 0403 are considered obsolete. Values 03e8 and 03eb are associated with Photoshop v2.0. The data format for values 03f2, 03f4-03fa, 03fc, 03fd, 0405-0bb7 is intentionally not documented by Adobe, or the data is missing. Please refer to the Adobe Photoshop SDK for information on obtaining the IPTC-NAA record 2 structure definition.

ID value 03ed indicates that the data is in the form of a ResolutionInfo structure:

```
typedef struct _ResolutionInfo
{
    LONG hRes;           /* Fixed-point number: pixels per inch */
    WORD hResUnit;       /* 1=pixels per inch, 2=pixels per centimeter */
    WORD WidthUnit;      /* 1=in, 2=cm, 3=pt, 4=picas, 5=columns */
    LONG vRes;           /* Fixed-point number: pixels per inch */
    WORD vResUnit;       /* 1=pixels per inch, 2=pixels per centimeter */
    WORD HeightUnit;     /* 1=in, 2=cm, 3=pt, 4=picas, 5=columns */
} RESOLUTIONINFO;
```

ID value 03ef indicates that the data is stored as a DisplayInfo structure, which contains display information associated with each channel:

```
typedef _DisplayInfo
{
    WORD ColorSpace;
    WORD Color[4];
    WORD Opacity;        /* 0-100 */
    BYTE Kind;           /* 0=selected, 1=protected */
    BYTE Padding;        /* Always zero */
} DISPLAYINFO;
```

ID value 03f3 indicates that the data is a series of eight flags, indicating the enabled state of labels, crop marks, color bars, registration marks, negative, flip, interpolate, and caption items in the Photoshop Page Setup dialog box.

ID value 2710 signals that the Data section contains a WORD-length version number (should be 1), a BYTE-length flag indicating crop marks, a BYTE-length field (should be 0), a LONG-length bleed width value, and a WORD indicating the bleed width scale.

## Layer and Mask Information Block

Following the Image Resources block is the Layer and Mask Information block, structured like the Color Mode Data and Image Resources blocks. In PSD files produced by Photoshop v2.5 the 4-byte Layer and Mask Information block length field contains a LONG number set to zero, as will vers3.0 files containing no layer or mask information. PSD file writers are advised that this is an area likely to be altered or expanded in future versions of the PSD format. Layer information in this block is stored first, followed by mask information.

### Layer Information

The Layer section of the Layer and Mask Information block starts with a LONG value specifying the length of the Layer Info section. This is followed by a WORD value count of the number of Layer Records to follow. Layer Records follow in sequence; a description of their organization is found below. Note that Layer Records vary in size from file to file depending on the number of channels in the image. There is one Layer Record for each layer in the image.

Data Structure	Field Name	Description
LONG	Top	Rectangle bounding the layer
LONG	Left	
LONG	Bottom	
LONG	Right	
WORD	Channels	Number of channels in the layer

The next area following contains a series of Channel Length Info records, defined as follows:

```
typedef struct _CLI
{
    WORD  ChannelID;           /* Channel Length Info field one */
    LONG  LengthOfChannelData; /* Channel Length Info field two */
} CLI;
```

Data Structure	Field Name	Description
CLI	Channel Length Info #1	
CLI	Channel Length Info #2	
.		
.		
.		
CLI	Channel Length Info # <i>n</i>	
BYTE[4]	Blend Mode Signature	Always "8BIM"
BYTE[4]	Blend Mode Key	(See table below)
BYTE	Opacity	0-255 (transparent to opaque)
BYTE	Clipping	0=base, 1=non-base
BYTE	Flags	Bit 0 = transparency protected, bit 1 = visible
BYTE	Padding	Set to zero
LONG	ExtraDataSize;	

The following area is the Layer Mask data section. If there is no Layer Mask, Size (the first LONG value) is set to zero, and the Layer Mask data section is four bytes long. If the value is non-zero, the Layer Mask data section is 24 bytes in length.

Data Structure	Field Name	Description
LONG	Size	
LONG	Top	Rectangle bounding layer mask
LONG	Left	
LONG	Bottom	
LONG	Right	
BYTE	DefaultColor	0 or 255
BYTE	Flags	Bit 0=position, bit 1=layer mask disabled, bit 2=invert layer mask
WORD	padding	Set to zero

The following area contains information defining the Layer Blending Ranges. This is followed by a series of records defining the source and destination ranges for each color channel. Each source and destination field consists of two black values (0-255) followed by two white values (0-255). Each channel source and destination range record has the following format:

```
typedef struct _CSDR
{
    BYTE Source[4]; /* First Channel Source Range */
    BYTE Dest[4];   /* First Channel Destination Range */
} CSDR;
```

Data Structure	Field Name	Description
LONG	Size	Total length of layer blending data
BYTE[4]	Gray Blend Source	
BYTE[4]	Gray Blend Destination	
CSDR	Channel Source Range #1	

CSDR	Channel Source Range #2	
CSDR	Channel Source Range # <i>n</i>	
BYTE[]	Layer Name	Pascal string, padded to multiple of 4 bytes

The Blend Mode Key field can contain the following ASCII keys:

**Key    Meaning**

norm	Normal
dark	Darken
lite	Lighten
hue	Hue
sat	Saturation
colr	Color
lum	Luminosity
mul	Multiply
scrn	Screen
diss	Dissolve
over	Overlay
hLit	Hard light
sLit	Soft light
diff	Difference

Following the list of Layer Records is the channel image data. Channel image data is stored in the form of 8-bit bitmaps. The first WORD value contains information about compression: 0 indicates that the data is uncompressed, and 1 indicates that the data is RLE compressed. Following this WORD is the actual channel image data. The length of the data is equal to the number of pixels in the image, which can be calculated from the bounding box. The data may be RLE compressed, however, using the PackBits algorithm described below. If the length of the channel image is odd, a pad byte is inserted to make the end of the image land on a WORD boundary.

**Mask Information**

Mask information consists of one or more mask info structures having the following format:

Data Structure	Field Name	Description
WORD	Overlay Color Space	(Not documented)
BYTE[8]	Color Components	4x2 byte color components
WORD	Opacity	0=transparent, 1=opaque
BYTE	Kind	0=inverted, 1=protected, 128=use stored value
BYTE	padding	set to zero

**Image Data**

After the Layer and Mask Information block is a WORD containing a value acting as a compression flag. If the flag is set (to 1), the image data is RLE compressed.

The image data is stored as indicated in the header and the compression value preceding it. If the data is compressed, each line is preceded by a WORD containing a value indicating the length, in bytes, of the data associated with that line. Lines are stored in scan-line order with no padding. Multiplanar data, which may or may not be compressed, is stored as a series of planes in sequence. That is, red data is stored first, followed by green data and then blue data. If the data is multiplanar, the WORD at the start of each line is the size, in bytes, of the total data associated with each scan-line (rows-in-line times number-of-channels).

Data is compressed using the Macintosh ROM PackBit encoding scheme, which is also part of the TIFF standard. The first byte of scan-line data read is a run count. If the MSB is set, the byte is converted to its two's complement value, and the next byte read is repeated that number of times. If the MSB is zero, one is added to the count, and the next RunCount bytes are read. In pseudocode, this is:

```
Zero BytesRead and Count
Read WORD LengthOfLineInBytes
Read a byte of data
Increment BytesRead
    If high bit is one
        Count is two's complement of byte
        Read next byte
        Write this value Count times
    If high bit is zero
        Count is byte value plus one
        Read next byte and write value Count times
Increment BytesRead
If BytesRead equals LengthOfLine the scan-line is done
```

Remember that LengthOfLineInBytes refers to all of the data associated with the current scan-line. If the image data is multiplanar, then BytesRead will have to be adjusted accordingly.

## Adobe Photoshop Raw File Format

The Photoshop Raw file format is used to import data into Photoshop from applications that cannot write file formats that are recognized by Photoshop. The Raw format is also used to export data from Photoshop that can be read by such applications.

The Raw format file is an uncompressed, binary file that contains only image data and no header or color palette information. The header data must be entered into Photoshop when the file is imported.

The Raw file data is nothing more than a series of pixel values starting from the upper-left pixel in the image and continuing downward. Each value is stored as a BYTE with a value in the range 0 (black, or least intense) to 255 (white, or most intense). RGB pixels are always stored in red-green-blue order, and CMYK pixels are always stored in cyan-magenta-yellow-key order.

To import a raw file, choose the Open item from the File menu, and click Show All Files. Choose Raw from the File pop-up menu, and click Open to display the Raw dialog box. Enter the width and height of the image in pixels (1 to 30,000), number of color channels (1 for gray-scale and palette color, 3 for RGB, 4 for CMYK, and so on), and the header size (the default is 0). Click OK.

When you export a raw file, only the image data is saved to the file, and all of the associated header and color palette data is lost. This data must be re-entered when importing the raw file. To export a raw file, choose Save As from the File menu and Raw from the Save As pop-up menu to display the Raw dialog box. Enter the File Type and File Creator values used to identify the Raw file, the header value (the default is 0), and the Interleaved Order (interleaved is the default). Click OK.

## For Further Information

The primary source of information on Photoshop is the following document, which Adobe has graciously allowed us to include.

Adobe Photoshop 3.0.4 Software Development Kit, Copyright 1991-1995, Adobe Systems
--

Some of the information for this article came from the following document, which is also included:

Adobe Photoshop 2.5 Format, Copyright 1992 and 1993, by Thomas Knoll and Adobe Systems, Inc.
--

For additional information, contact:

Adobe Systems  
Developer Support  
1585 Charleston Rd.  
P.O. Box 7900  
Mountain View CA 94039-7900  
Voice: 415-961-4400  
Voice: 800-344-8335  
FAX: 415-961-3769  
WWW: <http://www.adobe.com/>  
FTP: <ftp://ftp.adobe.com/>

This page is taken from the Encyclopedia of Graphics File Formats (/resource/book/1565921615/index.htm) and is licensed by O'Reilly (<https://www.oreilly.com/>) under the Creative Common/Attribution license.

[More Resources \(internal.htm\)](#)