Adobe Photoshop® 4.0

File Format Specification

Version 4.0.1 Release 1
April 1997

Most of the material for this document was derived from earlier works by Thomas Knoll, Mark Hamburg and Zalman Stern. Additional contributions came from David Corboy, Kevin Johnston, Sean Parent and Seetha Narayanan. It was then compiled and edited by Dave Wise and Paul Ferguson. It was later edited for content and updates by Andrew Coven.

Version History

Date	Author	Status
18 April1997	Andrew Coven	Extracted from SDK documentation for standalone document.

Table of Contents

1. Introduction

Welcome to the Adobe Photoshop® File Format Specification!

This document is the detailed specification of the Adobe Photoshop File Format, and any other pertinent file formats that Adobe Photoshop reads and writes.

Audience

This toolkit is for C programmers who wish to write plug–ins for Adobe Photoshop on Macintosh and Windows systems.

This guide assumes you are proficient in the C programming language and its tools. The source code files in this toolkit are written for Metrowerks CodeWarrior on the Macintosh, and Microsoft Visual C++ on Windows.

You should have a working knowledge of Adobe Photoshop, and understand how plug–in modules work from a user's viewpoint. This guide assumes you understand Photoshop terminology such as *paths*, *layers* and *masks*. For more information, consult the *Adobe Photoshop User Guide*.

This guide does not contain information on creating plug–in modules for Unix versions of Photoshop. The Photoshop Unix SDK is available on the Photoshop Unix product CD. You must purchase the product CD to obtain the SDK.

About this guide

This programmer's guide is designed for readability on screen as well as in printed form. The page dimensions were chosen with this in mind. The Frutiger and Minion font families are used throughout the manual.

To print this manual from within Adobe Acrobat Reader, select the "Shrink to Fit" option in the Print dialog.

GAP SDK tech notes mailing list

The Adobe Developers Association maintains an area on Adobe's world-wide-web site: `http://www.adobe.com`, which includes the latest SDK public releases and technical notes. You can also have the technical notes e–mailed to you directly by joining the Graphics and Publishing SDK tech notes mailing list. The GAP SDK Tech Notes e-mail list is for Adobe After Effects, Adobe Illustrator, Adobe PageMaker, Adobe Photoshop, Adobe PhotoDeluxe, and Adobe Premiere. To receive tech notes, send an e-mail to:

```
gap-sdk-tn-requests@adobe.com
```

with the subject:

```
SUBSCRIBE
```

and these fields in your message body:

1.   Your full name:

2.   Business name:

3.   Address:

4.   City:

5.   State:

6.   Country:

7.   Country code or Zip:

8.   Area code and phone number (business is fine):

9.   ADA member number:
      "N/A" if not a member; "Info" if want info.

SDK discussion mailing list

The Adobe Developers Association maintains an electronic mailing list that is used as peer discussion for developers. It is unmoderated and is populated by developers just like yourself, offering peer discussion of software development kit, Adobe plug-ins, and related issues. The mailing list is for discussion of all of the SDKs that fall under the ADA: Graphics and Publishing, which includes Adobe After Effects, Adobe Illustrator, Adobe PageMaker, Adobe Photoshop, Adobe PhotoDeluxe, and Adobe Premiere; Acrobat; FrameMaker; and PageMaker. To join the discussion send an e-mail to:

        sdk-requests@adobe.com

with the subject:

        SUBSCRIBE

and these fields in your message body:

1.   Your full name:

2.   Business name:

3.   Address:

4.   City:

5.   State:

6.   Country:

7.   Country code or Zip:

8.   Area code and phone number (business is fine):

9.   ADA member number:
      "N/A" if not a member; "Info" if want info.

2. Document File Formats
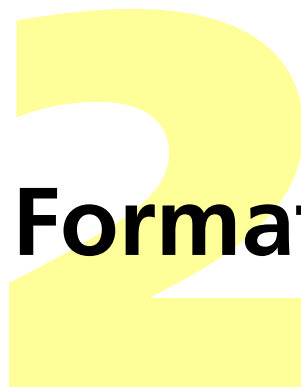
Adobe Photoshop saves a user's document in one of several formats, which are listed under the pop–up menu in the **Save** dialog. This chapter documents these standard formats.

The formats discussed in this chapter include Photoshop 3.0 native format, Photoshop 4.0 additions to the 3.0 file format, Photoshop EPS format, Filmstrip format, and TIFF format.

For more information about file formats, you may wish to consult the *Encyclopedia of Graphics File Formats* by James D. Murray & William vanRyper (1994, O'Reilly & Associates, Inc., Sebastopol, CA, ISBN 1–56592–058–9).

Image resource blocks

Image resource blocks are the basic building unit of several file formats, including Photoshop's native file format, JPEG, and TIFF. Image resources are used to store non–pixel data associated with an image, such as pen tool paths. (They are referred to as resource data because they hold data that was stored in the Macintosh's resource fork in early versions of Photoshop.)

The basic structure of Image Resource Blocks is shown in table .

Table 2–1: Image resource block

Type	Name	Description
OSType	Type	Photoshop always uses its signature, `8BIM`.
int16	ID	Unique identifier (see table 10–2).
PString	Name	A pascal string, padded to make size even (a null name consists of two bytes of 0)
int32	Size	Actual size of resource data. This does not include the `Type`, `ID`, `Name`, or `Size` fields.
Variable	Data	Resource data, padded to make size even

Image resources use several standard ID numbers, as shown in table 2–2. Not all file formats use all ID's. Some information may be stored in other sections of the file.

Table 2–2: Image resource IDs

ID Hex	Dec	Description
0x03E8	1000	Obsolete—Photoshop 2.0 only. Contains five `int16` values: number of channels, rows, columns, depth, and mode.
0x03E9	1001	Optional. Macintosh print manager print info record.
0x03EB	1003	Obsolete—Photoshop 2.0 only. Contains the indexed color table.
0x03ED	1005	`ResolutionInfo` structure. See Appendix A.
0x03EE	1006	Names of the alpha channels as a series of Pascal strings.
0x03EF	1007	`DisplayInfo` structure. See Appendix A .
0x03F0	1008	Optional. The caption as a Pascal string.
0x03F1	1009	Border information. Contains a fixed-number for the border width, and an int16 for border units (1=inches, 2=cm, 3=points, 4=picas, 5=columns).
0x03F2	1010	Background color. See the Colors file information in chapter 9.
0x03F3	1011	Print flags. A series of one byte boolean values (see Page Setup dialog): labels, crop marks, color bars, registration marks, negative, flip, interpolate, caption.
0x03F4	1012	Grayscale and multichannel halftoning information.
0x03F5	1013	Color halftoning information.
0x03F6	1014	Duotone halftoning information.
0x03F7	1015	Grayscale and multichannel transfer function.
0x03F8	1016	Color transfer functions.
0x03F9	1017	Duotone transfer functions.
0x03FA	1018	Duotone image information.
0x03FB	1019	Two bytes for the effective black and white values for the dot range.

Table 2–2: Image resource IDs (Continued)

ID		Description
Hex	**Dec**	
0x03FC	1020	Obsolete.
0x03FD	1021	EPS options.
0x03FE	1022	Quick Mask information. 2 bytes containing Quick Mask channel ID, 1 byte boolean indicating whether the mask was initially empty.
0x03FF	1023	Obsolete.
0x0400	1024	Layer state information. 2 bytes containing the index of target layer. 0=bottom layer.
0x0401	1025	Working path (not saved). See path resource format later in this chapter.
0x0402	1026	Layers group information. 2 bytes per layer containing a group ID for the dragging groups. Layers in a group have the same group ID.
0x0403	1027	Obsolete.
0x0404	1028	IPTC-NAA record. This contains the **File Info...** information.
0x0405	1029	Image mode for raw format files.
0x0406	1030	JPEG quality. Private.
0x0408	1032	*New since version 4.0 of Adobe Photoshop:* Grid and guides information. See grid and guides resource format later in this chapter. *New since version 4.0 of Adobe Photoshop.*
0x040A	1034	*New since version 4.0 of Adobe Photoshop:* Copyright flag. Boolean indicating whether image is copyrighted. Can be set via Property suite or by user in **File Info...**
0x040B	1035	*New since version 4.0 of Adobe Photoshop:* URL. Handle of a text string with uniform resource locator. Can be set via Property suite or by user in **File Info...**
0x07D0-0x0BB6	2000-2998	Path Information (saved paths). See path resource format later in this chapter.
0x0BB7	2999	Name of clipping path. See path resource format later in this chapter.
0x2710	10000	Print flags information. 2 bytes version (=1), 1 byte center crop marks, 1 byte (=0), 4 bytes bleed width value, 2 bytes bleed width scale.

Grid and guides resource format

Adobe Photoshop 4.0 and later stores grid and guides information an image in an image resource block. These resource blocks consist of an initial 12 byte grid and guide header, which is always present, followed by a 5 byte blocks of specific guide information for guide direction and location, which are present if there are guides (`fGuideCount` > 0).

Table 2–3: Grid and guide header

Type	Name	Description
int32	fVersion	=1 for Photoshop 4.0.
VPoint	fGridCycle	Future implementation of document-specific grids. Initially, set the grid cycle to every quarter inch. At 72 dpi, that would be 18 * 32 = 576 (0x240).
int32	fGuideCount	Can be 0. Otherwise, number of guide resource blocks.

Table 2–4: Guide resource block

Type	Name	Description
int32	fLocation	Location of guide in document coordinates. Since the guide is either vertical or horizontal, this only has to be one component of the coordinate.
VHSelect	fDirection	Direction of guide. VHSelect is a system type of unsigned char where 0 = vertical, 1 = horizontal.

Grid and guide information may be modified using the Property suite. See the Callbacks chapter for more information.

Path resource format

Photoshop stores the paths saved with an image in an image resource block. These resource blocks consist of a series of 26 byte path point records, and so the resource length should always be a multiple of 26.

Photoshop stores its paths as resources of type `8BIM` with IDs in the range 2000 through 2999. These numbers should be reserved for Photoshop. The name of the resource is the name given to the path when it was saved.

If the file contains a resource of type `8BIM` with an ID of 2999, then this resource contains a Pascal–style string containing the name of the clipping path to use with this image when saving it as an EPS file.

The path format returned by `GetProperty()` call is identical to what is described below. Refer to the `IllustratorExport` sample plug–in code to see how this resource data is constructed.

Path points

All points used in defining a path are stored in eight bytes as a pair of 32–bit components, vertical component first.

The two components are signed, fixed point numbers with 8 bits before the binary point and 24 bits after the binary point. Three guard bits are reserved in the points to eliminate most concerns over arithmetic overflow. Hence, the range for each component is `0xF0000000` to `0x0FFFFFFF` representing a range of –16 to 16. The lower bound is included, but not the upper bound.

This limited range is used because the points are expressed relative to the image size. The vertical component is given with respect to the image height, and the horizontal component is given with respect to the image width. [0,0] represents the top–left corner of the image; [1,1] ([`0x01000000`,`0x01000000`]) represents the bottom–right.

In Windows, the byte order of the path point components are reversed; you should swap the bytes when accessing each 32–bit value.

Path records

The data in a path resource consists of one or more 26-byte records. The first two bytes of each record is a selector to indicate what kind of path it is. For Windows, you should swap the bytes before accessing it as a short (`int16`).

Table 2–5: Path data record types

Selector	Description
0	Closed subpath length record
1	Closed subpath Bezier knot, linked
2	Closed subpath Bezier knot, unlinked
3	Open subpath length record
4	Open subpath Bezier knot, linked
5	Open subpath Bezier knot, unlinked
6	Path fill rule record
7	Clipboard record

The first 26-byte path record contains a selector value of 6, path fill rule record. The remaining 24 bytes of the first record are zeroes. Paths use even/

odd ruling. Subpath length records, selector value 0 or 3, contain the number of Bezier knot records in bytes 2 and 3. The remaining 22 bytes are unused, and should be zeroes. Each length record is then immediately followed by the Bezier knot records describing the knots of the subpath.

In Bezier knot records, the 24 bytes following the selector field contain three path points (described above) for:

1.   the control point for the Bezier segment preceding the knot,

2.   the anchor point for the knot, and

3.   the control point for the Bezier segment leaving the knot.

Linked knots have their control points linked. Editing one point modifies the other to preserve collinearity. Knots should only be marked as having linked controls if their control points are collinear with their anchor. The control points on unlinked knots are independent of each other. Refer to the *Adobe Photoshop User Guide* for more information.

Clipboard records, `selector=7`, contain four fixed-point numbers for the bounding rectangle (top, left, bottom, right), and a single fixed-point number indicating the resolution.

Photoshop 3.0 files

This is the native file format for Adobe Photoshop 3.0. It supports storing all layer information.

Table 2–6: Photoshop 3.0 file types

OS	Filetype/extension
Mac OS	`8BPS`
Windows	`.PSD`

Photoshop 3.0 files under Windows

All data is stored in big endian byte order; under Windows you must byte swap short and long integers when reading or writing.

Photoshop 3.0 files under Mac OS

For cross–platform compatibility, all information needed by Adobe Photoshop 3.0 is stored in the data fork. For interoperability with other Macintosh applications, however, some information is duplicated in resources stored in the resource fork of the file.

For compatibility with image cataloging applications, the `pnot` resource id 0 contains references to thumbnail, keywords, and caption information stored in other resources. The thumbnail picture is stored in a `'PICT'` resource, the keywords are stored in `'STR#'` resource 128 and the caption text is stored in `'TEXT'` resource 128. For more information on the format of these resources see *Inside Macintosh: QuickTime Components* and the *Extensis Fetch Awareness Developer's Toolkit.*

Photoshop also creates `'icl8'` –16455 and `'ICN#'` –16455 resources containing thumbnail images which will be shown in the Mac OS Finder.

All of the data from Photoshop's File Info dialog is stored in `'ANPA'` resource 10000. The data in this resource is stored as an IPTC–NAA record 2 and should be readable by various tools from Iron Mike. For more information on the format of this resource contact:

IPTC–NAA Digital Newsphoto Parameter Record
Newspaper Association of America
The Newspaper Center
11600 Sunrise Valley Drive
Reston VA 20091

Photoshop 3.0 file format

The file format for Photoshop 3.0 is divided into five major parts.

| File Header |
| Color Mode Data |
| Image Resources |
| Layer and Mask Information |
| Image Data |
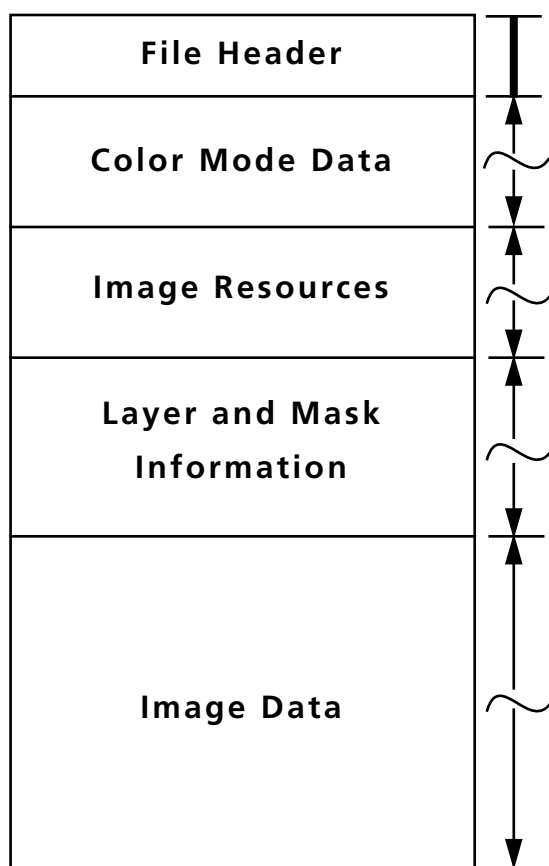
The file header is fixed length, the other four sections are variable in length.

When writing one of these sections, you should write all fields in the section, as Photoshop may try to read the entire section. Whenever writing a file and skipping bytes, you should explicitly write zeros for the skipped fields.

When reading one of the length delimited sections, use the length field to decide when you should stop reading. In most cases, the length field indicates the number of bytes, not records, following.

File header section
The file header contains the basic properties of the image.

Table 2–7: File header

Length	Name	Description
4 bytes	Signature	Always equal to 8BPS. Do not try to read the file if the signature does not match this value.
2 bytes	Version	Always equal to 1. Do not try to read the file if the version does not match this value.
6 bytes	Reserved	Must be zero.
2 bytes	Channels	The number of channels in the image, including any alpha channels. Supported range is 1 to 24.
4 bytes	Rows	The height of the image in pixels. Supported range is 1 to 30,000.
4 bytes	Columns	The width of the image in pixels. Supported range is 1 to 30,000.
2 bytes	Depth	The number of bits per channel. Supported values are 1, 8, and 16.
2 bytes	Mode	The color mode of the file. Supported values are: Bitmap=0; Grayscale=1; Indexed=2; RGB=3; CMYK=4; Multichannel=7; Duotone=8; Lab=9.

Color mode data section

Only indexed color and duotone have color mode data. For all other modes, this section is just 4 bytes: the length field, which is set to zero.

For indexed color images, the length will be equal to 768, and the color data will contain the color table for the image, in non–interleaved order.

For duotone images, the color data will contain the duotone specification, the format of which is not documented. Other applications that read Photoshop files can treat a duotone image as a grayscale image, and just preserve the contents of the duotone information when reading and writing the file.

Table 2–8: Color mode data

Length	Name	Description
4 bytes	Length	The length of the following color data.
Variable	Color data	The color data.

Image resources section

The third section of the file contains image resources. As with the color mode data, the section is indicated by a length field followed by the data. The image resources in this data area are described in detail earlier in this chapter.

Table 2–9: Image resources

Length	Name	Description
4 bytes	Length	Length of image resource section.
Variable	Resources	Image resources.

Layer and mask information section

The fourth section contains information about Photoshop 3.0 layers and masks. The formats of these records are discussed later in this chapter. If there are no layers or masks, this section is just 4 bytes: the length field, which is set to zero.

Table 2–10: Layer and mask information

Length	Name	Description
4 bytes	Length	Length of the miscellaneous information section.
Variable	Layers	Layer info. See table 2–12.
Variable	Global layer mask	Global layer mask info. See table 2–19.

Image data section

The image pixel data is the last section of a Photoshop 3.0 file. Image data is stored in planar order, first all the red data, then all the green data, etc. Each plane is stored in scanline order, with no pad bytes.

If the compression code is 0, the image data is just the raw image data.

If the compression code is 1, the image data starts with the byte counts for all the scan lines (rows * channels), with each count stored as a two–byte value. The RLE compressed data follows, with each scan line compressed separately. The RLE compression is the same compression algorithm used by the Macintosh ROM routine `PackBits`, and the TIFF standard.

Table 2–11: Image data

Length	Name	Description
2 bytes	Compression	Compression method. Raw data = 0, RLE compressed = 1.
Variable	Data	The image data. Planar order = RRR GGG BBB, etc.

Layer and mask records

Information about each layer and mask in a document is stored in the fourth section of the file. The complete, merged image data is not stored here; it resides in the last section of the file.

The first part of this section of the file contains layer information, which is divided into layer structures and layer pixel data, as shown in table 2–12. The second part of this section contains layer mask data, which is described in table 2–19.

Table 2–12: Layer info section

Length	Name	Description
4 bytes	Length	Length of the layers info section, rounded up to a multiple of 2.
Variable	Layers structure	Data about each layer in the document. See table 2–13.
Variable	Pixel data	Channel image data for each channel in the order listed in the layers structure section. See table 2–18.

Table 2–13: Layer structure

Length	Name	Description
2 bytes	Count	Number of layers. If <0, then number of layers is absolute value, and the first alpha channel contains the transparency data for the merged result.
Variable	Layer	Information about each layer (table 2–18).

Table 2–14: Layer records

Length	Name	Description
4 bytes	Layer top	The rectangle containing the contents of the layer.
4 bytes	Layer left	
4 bytes	Layer bottom	
4 bytes	Layer right	
2 bytes	Number channels	The number of channels in the layer.
Variable	Channel length info	Channel information. This contains a six byte record for each channel. See table 2–15.
4 bytes	Blend mode signature	Always `8BIM`.
4 bytes	Blend mode key	`'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
1 byte	Opacity	0 = transparent ... 255 = opaque
1 byte	Clipping	0 = base, 1 = non–base

Table 2–14: Layer records (Continued)

Length	Name	Description
1 byte	Flags	bit 0 = transparency protected bit 1 = visible
1 byte	(filler)	(zero)
4 bytes	Extra data size	Length of the extra data field. This is the total length of the next five fields.
24 bytes, or 4 bytes if no layer mask.	Layer mask data	See table 2–16.
Variable	Layer blending ranges	See table 2–17.
Variable	Layer name	Pascal string, padded to a multiple of 4 bytes.
These fields are new since version 4.0 of Adobe Photoshop:		
Variable	Adjustment layer info	See table 2–20.

Table 2–15: Channel length info

Length	Name	Description
2 bytes	Channel ID	0 = red, 1 = green, etc. −1 = transparency mask −2 = user supplied layer mask
4 bytes	Length	Length of following channel data.

Table 2–16: Layer mask / adjustment layer data

Length	Name	Description
4 bytes	Size	Size of the data. This will be either 0x14, or zero (in which case the following fields are not present).
4 bytes	Top	Rectangle enclosing layer mask.
4 bytes	Left	
4 bytes	Bottom	
4 bytes	Right	
1 byte	Default color	0 or 255
1 byte	Flags	bit 0 = position relative to layer bit 1 = layer mask disabled bit 2 = invert layer mask when blending
2 bytes	Padding	Zeros

Table 2–17: Layer blending ranges data

Length	Name	Description
4 bytes	Length	Length of layer blending ranges data
4 bytes	Composite gray blend source	Contains 2 black values followed by 2 white values. Present but irrelevant for Lab & Grayscale.
4 bytes	Composite gray blend destination	Destination Range
4 bytes	First channel source range	First channel source

Table 2–17: Layer blending ranges data (Continued)

Length	Name	Description
4 bytes	First channel destination range	First channel destination
4 bytes	Second channel source range	Second channel source
4 bytes	Second channel destination range	Second channel destination
...	...	...
4 bytes	Nth channel source range	Nth channel source
4 bytes	Nth channel destination range	Nth channel destination

Table 2–18: Channel image data

Length	Name	Description
2 bytes	Compression	0 = Raw Data, 1 = RLE compressed.
Variable	Image data	If the compression code is 0, the image data is just the raw image data calculated as `((LayerBottom-LayerTop)*(LayerRight-LayerLeft))`. If the compression code is 1, the image data starts with the byte counts for all the scan lines in the channel `(LayerBottom-LayerTop)`, with each count stored as a two–byte value. The RLE compressed data follows, with each scan line compressed separately. The RLE compression is the same compression algorithm used by the Macintosh ROM routine PackBits, and the TIFF standard. If the Layer's Size, and therefore the data, is odd, a pad byte will be inserted at the end of the row. *New since version 4.0 of Adobe Photoshop:* If the layer is an adjustment layer, the channel data is undefined (probably all white.)

Table 2–19: Global layer mask info

Length	Name	Description
4 bytes	Length	Length of global layer mask info section.
2 bytes	Overlay color space	Overlay color space (undocumented).
8 bytes	Color components	4 * 2 byte color components
2 bytes	Opacity	0 = transparent, 100 = opaque.
1 byte	Kind	0=Color selected—i.e. inverted; 1=Color protected;128=use value stored per layer. This value is preferred. The others are for backward compatibility with beta versions.
1 byte	(filler)	(zero)

Photoshop 4.0 file format

The Photoshop 4.0 file format is an extension of the Photoshop 3.0 file format. It is essentially the same, with some additional image resources and resource blocks. Additional resources for 4.0 are:

1. Copyright flag. Image resource ID 1034 (0x040A). See table 2–2

2. URL ID. Image resource ID 1035 (0x040B). See table 2–2.

3. Grid and guides information. See table 2–2 and grid and guides resource format, earlier in this chapter.

4. Adjustment layers. See table 2–14 and 2–20.

Table 2–20: Adjustment layer info

Length	Name	Description
4 bytes	signature	always '8BIM'
4 bytes	key	OSType key for which adjustment type to use: 'levl'=Levels 'curv'=Curves 'brit'=Brightness/contrast 'blnc'=Color balance 'hue '=Hue/saturation 'selc'=Selective color 'thrs'=Threshold 'nvrt'=Invert 'post'=Posterize
4 bytes	length	Length of adjustment data, below.
Variable	data	Adjustment data. Same as load file formats for each format. See Load File Formats chapter for information.

Photoshop EPS files

Photoshop 3.0 and later writes a high–resolution bounding box comment to the EPS file immediately following the traditional EPS bounding box comment. The comment begins with "`%%HiResBoundingBox`" and is followed by four numbers identical to those given for the bounding box except that they can have fractional components (i.e., a decimal point and digits after it). The traditional bounding box is written as the rounded version of the high resolution bounding box for compatibility.

Photoshop writes its image resources out to a block of data stored as follows:

```
%BeginPhotoshop: <length> <hex data>
```

Table 2–21: EPS parameters for BeginPhotoshop

Field	Definition
length	Length of the image resource data.
hex data	Image resource data in hexadecimal.

Photoshop includes a comment in the EPS files it writes so that it is able to read them back in again. Third party programs that write pixel–based EPS files may want to include this comment in their EPS files, so Photoshop can read their files.

The comment must follow immediately after the `%%` comment block at the start of the file. The comment is:

```
%ImageData: <columns> <rows> <depth> <mode> <pad channels> <block size>
<binary/hex> "<data start>"
```

Table 2–22: EPS parameters for ImageData

Field	Definition
columns	Width of the image in pixels.
rows	Height of the image in pixels.
depth	Number of bits per channel. Must be 1 or 8.
mode	Image mode. Bitmap/grayscale=1; Lab=2; RGB=3; CMYK=4.
pad channels	Number of other channels store in the file. Ignored when reading. Photo-shop uses this to include a grayscale image that is printed on non-color PostScript printers.
block size	Number of bytes per row per channel. Will be either 1 or formula (below): 1=Data is interleaved. `(columns*depth+7)/8`=Data is stored in line-interleaved format, or there is only one channel.
binary/ascii	1=Data is in binary format. 2=Data is in hex ascii format.
data start	Entire PostScript line immediately preceding the image data. This entire line should not occur elsewhere in the PostScript header code, but it may occur at part of a line.

Filmstrip files

Adobe Premiere 2.0 and later supports the filmstrip file format. Premiere users can export any video clip as a filmstrip. Refer to the *Adobe Premiere User Guide* for more information.

Adobe Photoshop 3.0 supports the filmstrip file type to allow each frame to be individually painted. The filmstrip file format is fairly simple, and is described in this section.

A filmstrip consists of a sequence of equal sized 32–bit images, known as frames. The channel order in the file is Red, Green, Blue, Alpha.

After each frame is an arbitrarily sized leader area, in which any type of information may be embedded. Adobe Premiere puts the timecode and frame number for the frame in this area. This area is ignored by Photoshop when the file is read.

Following all the frames is a 16 row trailer frame (it has the same width as the other frames). Adobe Premiere writes a yellow and black diagonal pattern in this area. The lower right corner of this area is actually an information record that exists at the very end of the file. This record is located by seeking to the end of the file minus the size of the record, then reading the record and verifying the signature field that it contains.

```
// Definition for filmstrip info record

typedef struct {
    long                signature;   // 'Rand'
    long                numFrames;   // number of frames in file
    short               packing;     // packing method
    short               reserved;    // reserved, should be 0
    short               width;       // image width
    short               height;      // image height
    short               leading;     // horiz gap between frames
    short               framesPerSec;// frame rate
    char                spare[16];   // some spare data.
} FilmStripRec, **FilmStripHand;
```

Table 2–23: FilmStripRec structure

Type	Field	Description
long	signature	This field must be set to the code `Rand` and is used to verify the validity of the record.
long	numFrames	This is the total number of frames in the file.
short	packing	This is the packing method used, currently only a value of 0 is defined, for no packing.
short	width	The width of each image, in pixels.
short	height	The height of each image, in pixels.
short	leading	The height of the leading areas, in pixels.
short	framesPerSec	The rate at which the frames should be played.

To locate the filmstrip info record, seek to the end of the file minus (`sizeof(FilmStripRec)`), then read in the FilmStrip record. Check the signature field for the code `Rand` to test for validity.

To locate the data for a particular frame, seek to

```
(frame * width * (height+leading) * 4)
```

then read the number of bytes in

```
(width * height * 4).
```

If the data is being placed into a Mac OS GWorld, the channels must be re–arranged from Red–Green–Blue–Alpha to Alpha–Red–Green–Blue.

To write a FilmStrip file, write each frame sequentially into the file, including the leading areas.

Then write this block of bytes:

```
((width * (height+leading) * 4) – sizeof(FilmStripRec)).
```

Finally, fill in and write the FilmStrip record to the file.

Note: The packing field should currently be zero. In the future packing methods may be defined for filmstrips, so any software which reads filmstrips should examine this field before opening the file.

TIFF files

The same image resources information found in Photoshop 3.0 files are stored in TIFF files under tag number 34377 (see Image Resource Blocks and Image Resources earlier in this chapter).

For TIFF files the caption data is stored in an image description tag 270 and all the information is stored as an IPTC–NAA record 2 in tag 33723. The tag number was chosen by inspecting files written by Iron Mike software, and is supposed to be defined in a Rich TIFF specification. The tag is also specified in:

NSK TIFF

> The Japan Newspaper Publishers & Editors Association
> Nippon Press Center Building
> 2–2–1 Uchlsaiwai–cho
> Chiyoda–ku, Tokyo 100

For more information about the TIFF format see:

TIFF Revision 6.0

```
http: //www.adobe.com/supportservice/devrelations/
       resources.html#tiff
```

In reading the files, the following order is used with information read lower on the list replacing information read higher:
> Image Description Tag (TIFF only)
> IPTC–NAA Tag (TIFF only)

Note: It is a bug that the TIFF information comes prior to the image resource information on this list. This means that an edit to the TIFF info will not be recognized unless the image resource information is removed. The TIFF data may be moved to after the image resource information in a future version of Photoshop.

Table 2–24 describes the standard TIFF tags and tag values that Photoshop 3.0 and later is able to read and write.

TIFF files under the Mac OS

For cross–platform compatibility, all TIFF information is stored in the data fork. For interoperability with other Macintosh applications, however, some information is duplicated in resources stored in the resource fork of the file.

For compatibility with image cataloging applications, the `pnot` resource id 0 contains references to thumbnail, keywords, and caption information stored in other resources. The thumbnail picture is stored in a `'PICT'` resource, the keywords are stored in `'STR#'` resource 128 and the caption text is stored in `'TEXT'` resource 128. For more information on the format of these resources see *Inside Macintosh: QuickTime Components* and the *Extensis Fetch Awareness Developer's Toolkit.*

All of the data from Photoshop's File Info dialog is stored in `'ANPA'` resource 10000. The TIFF file also contains `'STR '` resource -16396 indicating the application that created the TIFF file. The string is "Adobe Photoshop™ 3.0" for Photoshop 3.0 and "Adobe Photoshop® 4.0" for Photoshop 4.0.

Photoshop also creates `'icl8'` –16455 and `'ICN#'` –16455 resources containing thumbnail images which will be shown in the Mac OS Finder.

Table 2–24: TIFF Tags

Tag	Photoshop reads	Photoshop writes
IFD	First IFD in file	Only one IFD per file
NewSubFileType	Ignored	0
ImageWidth	1 to 30000	1 to 30000
ImageLength	1 to 30000	1 to 30000
BitsPerSample	1, 2, 4, 8, 16 (all same)	1, 8, 16
Compression	1, 2, 5, 32773	1, 5
PhotometricInterpretation	0, 1, 2, 3, 5, 8	0 (1–bit), 1 (8–bit), 2, 3,5,8
FillOrder	1	No
ImageDescription	Printing Caption	Printing Caption
StripOffsets	Yes	Yes
SamplesPerPixel	1 to 24	1 to 24
RowsPerStrip	Any	Single strip if not compressed, multiple strips if compressed.
StripByteCounts	Required if compressed	Yes
XResolution	Yes	Yes
YResolution	Ignored (square pixels assumed)	Yes
PlanarConfiguration	1 or 2	1
ResolutionUnit	2 or 3	2
Predictor	1 or 2	1 or 2
ColorMap	Yes	Yes
TileWidth	Yes	No
TileLength	Yes	No
TileOffsets	Yes	No
TileByteCounts	Required if compressed	No
InkSet	1	No
DotRange	Yes, if CMYK	Yes
ExtraSamples	Ignored (except for count)	0

Numerics

A

B

C

D

E

F

G

GAP SDK tech notes mailing list 4
GetProperty() 10
Grayscale 7

H

halftoning 7
height 21
hue 19
Hue/saturation 19

I

icl8 12, 23
ICN# 12, 23
IFD 24
ImageDescription 24
ImageLength 24
ImageWidth 24
InkSet 24
Invert 19
invt 19
IPTC-NAA 8
IPTC–NAA 23

J

JPEG 8

K

Kind 18

L

Layer 16
Layer blending ranges 17
Layer mask data 17
Layer name 17
Layer top 16
Layers 14
Layers structure 16
leading 21
Length 14
Levels 19
levl 19

M

Masks 14
Mode 13
mode 20

N

NewSubFileType 24
numFrames 21

O

Opacity 16, 18
Overlay color space 18

P

packing 21