



Adobe

# Adobe Photoshop® 6.0



## File Formats Specification

Version 6.0 Release 2  
November 2000

## Adobe Photoshop File Formats Specification

Copyright © 1991–2000 Adobe Systems Incorporated.

All rights reserved.

Portions Copyright © 1990–1991, Thomas Knoll.

The information in this document is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Adobe Systems Incorporated. Adobe Systems Incorporated assumes no responsibility or liability for any errors or inaccuracies that may appear in this document. The software described in this document is furnished under license and may only be used or copied in accordance with the terms of such license.

Adobe, Adobe After Effects, Adobe PhotoDeluxe, Adobe Premiere, Adobe Photoshop, Adobe Illustrator, Adobe Type Manager, ATM and PostScript are trademarks of Adobe Systems Incorporated that may be registered in certain jurisdictions. Macintosh and Apple are registered trademarks, and Mac OS is a trademark of Apple Computer, Inc. Microsoft, Windows and Windows95 are registered trademarks of Microsoft Corporation. All other products or name brands are trademarks of their respective holders.

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 and Thomas Ruark.

## Version History

Date	Author	Status
18 April 1997	Andrew Coven	Extracted from SDK documentation for standalone document.
1 June 1998	Andrew Coven	Updated for Photoshop 5.0.
1 October 1998	Thomas Ruark	Image Resources in 5.0 format.
25 August 1999	Thomas Ruark	Type Tool info clean up.
10 August 2000	Thomas Ruark	Update for Photoshop 6.0
14 November 2000	Thomas Ruark	Things missed in 6.0

<b>Title Page</b> . . . . .	<b>1</b>
Version History . . . . .	2
<b>Table of Contents</b> . . . . .	<b>3</b>
<b>1. Introduction</b> . . . . .	<b>5</b>
Audience . . . . .	5
About this guide . . . . .	5
SDK discussion mailing list . . . . .	6
<b>2. Document File Formats</b> . . . . .	<b>7</b>
Image resource blocks. . . . .	8
Grid and guides resource format . . . . .	11
Thumbnail resource format . . . . .	12
Color Samplers resource format. . . . .	13
Path resource format . . . . .	14
Path points . . . . .	14
Path records. . . . .	14
Slices resource format. . . . .	16
Photoshop 3.0 files . . . . .	18
Photoshop 3.0 files under Windows . . . . .	18
Photoshop 3.0 files under Mac OS . . . . .	18
Photoshop 3.0 file format. . . . .	19
File header section . . . . .	19
Color mode data section . . . . .	20
Image resources section . . . . .	20
Layer and mask information section . . . . .	20
Image data section . . . . .	21
Layer and mask records . . . . .	22
Photoshop 4.0 file format. . . . .	26
Photoshop 5.0 file format. . . . .	27
Photoshop 6.0 file format. . . . .	31
Photoshop EPS files. . . . .	39
Filmstrip files . . . . .	40
TIFF files . . . . .	42
TIFF files under the Mac OS . . . . .	42
<b>3. Additional File Formats</b> . . . . .	<b>44</b>
Actions . . . . .	45
Arbitrary Map . . . . .	49
Brushes. . . . .	50
CMYK Setup. . . . .	52
Color Table. . . . .	53
Colors . . . . .	54
Command Settings File. . . . .	56

Curves . . . . .	58
Custom kernel . . . . .	60
Duotone options . . . . .	61
Halftone screens . . . . .	63
Hue/Saturation . . . . .	64
Ink colors setup . . . . .	66
Levels . . . . .	67
Monitor setup . . . . .	69
Replace color/Color range . . . . .	70
Scratch Area . . . . .	71
Selective color . . . . .	72
Separation setup . . . . .	73
Separation tables . . . . .	74
Transfer function . . . . .	76
. . . . .	77
<b>Index . . . . .</b>	<b>78</b>

# 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.

## 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.

To un-join the discussion send the same email with UNSUBSCRIBE in the subject line.

# 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 6.0, 5.0, and 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 2-1.

**Table 2-1: Image resource block**

Type	Name	Description
OSType	Type	Photoshop always uses its signature, 8BIM.
2 bytes	ID	Unique identifier. <a href="#">Image resource IDs on page 8</a>
PString	Name	A pascal string, padded to make size even (a null name consists of two bytes of 0)
4 bytes	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		Description
Hex	Dec	
0x03E8	1000	Obsolete—Photoshop 2.0 only. Contains five 2 byte 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 in <i>Photoshop SDK Guide.pdf</i> .
0x03EE	1006	Names of the alpha channels as a series of Pascal strings.
0x03EF	1007	DisplayInfo structure. See Appendix A in <i>Photoshop SDK Guide.pdf</i> .
0x03F0	1008	Optional. The caption as a Pascal string.
0x03F1	1009	Border information. Contains a fixed-number for the border width, and 2 bytes for border units (1=inches, 2=cm, 3=points, 4=picas, 5=columns).
0x03F2	1010	Background color. See the Colors additional file information.
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.



Table 2–2: Image resource IDs (Continued)

ID		Description
Hex	Dec	
0x03FB	1019	Two bytes for the effective black and white values for the dot range.
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. See the IIMV4.pdf document.
0x0405	1029	Image mode for raw format files.
0x0406	1030	JPEG quality. Private.
0x0408	1032	Grid and guides information. See grid and guides resource format later in this chapter.
0x0409	1033	Thumbnail resource. See thumbnail resource format later in this chapter.
0x040A	1034	Copyright flag. Boolean indicating whether image is copyrighted. Can be set via Property suite or by user in File Info...
0x040B	1035	URL. Handle of a text string with uniform resource locator. Can be set via Property suite or by user in File Info...
0x040C	1036	Thumbnail resource. See thumbnail resource format later in this chapter.
0x040D	1037	Global Angle. 4 bytes that contain an integer between 0..359 which is the global lighting angle for effects layer. If not present assumes 30.
0x040E	1038	Color samplers resource. See color samplers resource format later in this chapter.
0x040F	1039	ICC Profile. The raw bytes of an ICC format profile, see the ICC34.pdf and ICC34.h files from the International Color Consortium located in the documentation section.
0x0410	1040	One byte for Watermark.
0x0411	1041	ICC Untagged. 1 byte that disables any assumed profile handling when opening the file. 1 = intentionally untagged.
0x0412	1042	Effects visible. 1 byte global flag to show/hide all the effects layer. Only present when they are hidden.
0x0413	1043	Spot Halftone. 4 bytes for version, 4 bytes for length, and the variable length data.
0x0414	1044	Document specific IDs, layer IDs will be generated starting at this base value or a greater value if we find existing IDs to already exceed it. It's purpose is to avoid the case where we add layers, flatten, save, open, and then add more layers that end up with the same IDs as the first set. 4 bytes.
0x0415	1045	Unicode Alpha Names. 4 bytes for length and the string as a unicode string.

Table 2–2: Image resource IDs (Continued)

ID		Description
Hex	Dec	
0x0416	1046	<i>New since version 6.0 of Adobe Photoshop:</i> Indexed Color Table Count. 2 bytes for the number of colors in table that are actually defined
0x0417	1047	<i>New since version 6.0 of Adobe Photoshop:</i> Transparent Index. 2 bytes for the index of transparent color, if any.
0x0419	1049	<i>New since version 6.0 of Adobe Photoshop:</i> Global Altitude. 4 byte entry for altitude
0x041A	1050	<i>New since version 6.0 of Adobe Photoshop:</i> Slices. See description later in this chapter
0x041B	1051	<i>New since version 6.0 of Adobe Photoshop:</i> Workflow URL. Unicode string, 4 bytes of length followed by unicode string.
0x041C	1052	<i>New since version 6.0 of Adobe Photoshop:</i> Jump To XPEP. 2 bytes major version, 2 bytes minor version, 4 bytes count. Following is repeated for count: 4 bytes block size, 4 bytes key, if key = 'jtDd' then next is a Boolean for the dirty flag otherwise it's a 4 byte entry for the mod date.
0x041D	1053	<i>New since version 6.0 of Adobe Photoshop:</i> Alpha Identifiers. 4 bytes of length, followed by 4 bytes each for every alpha identifier.
0x041E	1054	<i>New since version 6.0 of Adobe Photoshop:</i> URL List. 4 byte count of URLs, followed by 4 byte long, 4 byte ID, and unicode string for each count.
0x0421	1057	<i>New since version 6.0 of Adobe Photoshop:</i> Version Info. 4 byte version, 1 byte HasRealMergedData, unicode string of writer name, unicode string of reader name, 4 bytes of file version.
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 for 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
4 bytes	fVersion	=1 for Photoshop 4.0.
8 bytes 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).
4 bytes	fGuideCount	Can be 0. Otherwise, number of guide resource blocks.

**Table 2–4: Guide resource block**

Type	Name	Description
4 bytes	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.
1 byte 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 in *Photoshop SDK Guide.pdf* for more information.

## Thumbnail resource format

Adobe Photoshop 5.0 and later stores thumbnail information for preview display in an image resource block. These resource blocks consist of an initial 28 byte header, followed by a JFIF thumbnail in RGB (red, green, blue) order for both Macintosh and Windows. Adobe Photoshop 4.0 stored the thumbnail information in the same format except the data section is BGR (blue, green, red). The Adobe Photoshop 4.0 format is at resource ID 1033 and the Adobe Photoshop 5.0 format is at resource ID 1036.

**Table 2–5: Thumbnail resource header**

Type	Name	Description
4 bytes	format	= 1 (kJpegRGB). Also supports kRawRGB (0).
4 bytes	width	Width of thumbnail in pixels.
4 bytes	height	Height of thumbnail in pixels.
4 bytes	widthbytes	Padded row bytes as $(width * bitspixel + 31) / 32 * 4$ .
4 bytes	size	Total size as $widthbytes * height * planes$
4 bytes	compressedsize	Size after compression. Used for consistency check.
2 bytes	bitspixel	= 24. Bits per pixel.
2 bytes	planes	= 1. Number of planes.
Variable	Data	JFIF data in RGB format. Note: For resource ID 1033 the data is in BGR format.

## Color Samplers resource format

Adobe Photoshop 5.0 and later stores color samplers information for an image in an image resource block. These resource blocks consist of an initial 8 byte color samplers header followed by a variable length block of specific color samplers information.

**Table 2–6: Color Samplers header**

Type	Name	Description
4 bytes	version	= 1 for Photoshop 5.0.
4 bytes	count	Number of color samplers to follow. See Color Samplers resource block in the next table.

**Table 2–7: Color Samplers resource block**

Type	Name	Description
VPoint	Position	The vertical and horizontal position of the point. Location = pixel position * 32 + 16.
2 bytes	Color Space	enum{ kDummySpace = -1, kRGBSpace, kHSBspace, kCMYKSpace, kPantoneSpace, kFocoltoneSpace, kTrumatchSpace, kToyoSpace, kLabSpace, kGraySpace, kWideCMYKSpace, kHKSSpace, kDICSpace, kTotalInkSpace, kMonitorRGBSpace, kDuotoneSpace, kOpacitySpace};

## 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.

**Table 2-8: 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
8	Initial fill rule 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.

Initial fill records, `selector=8`, contain one two byte record. A value of 1 means that the fill starts with all pixels. The value will be either 0 or 1.

## Slices resource format

Adobe Photoshop 6.0 and later stores slices information for an image in an image resource block. .

**Table 2–9: Slices header**

Type	Name	Description
4 bytes	version	=6 for Photoshop 6.0.
4 * 4 bytes	rect	Bounding rect for all of the slices: top, left, bottom, right of all the slices
4 bytes	length	length of unicode string to follow
Variant	unicode string	unicode string, not present if length = 0
4 bytes	count	Number of slices to follow. See Slices resource block in the next table.

**Table 2–10: Slices resource block**

Type	Name	Description
4 bytes	ID	ID
4 bytes	GroupID	group ID
4 bytes	origin	origin
4 bytes	length	length of unicode string to follow
Variant	unicode string	unicode string, not present if length = 0
4 bytes	type	type
4 bytes	left	left position
4 bytes	top	top position
4 bytes	right	right position
4 bytes	bottom	bottom position
4 bytes	length	length of unicode string to follow
Variant	URL	unicode string, not present if length = 0
4 bytes	length	length of unicode string to follow
Variant	Target	unicode string, not present if length = 0
4 bytes	length	length of unicode string to follow
Variant	Message	unicode string, not present if length = 0
4 bytes	length	length of unicode string to follow
Variant	Alt Tag	unicode string, not present if length = 0
1 byte	Boolean	cell text is HTML
4 bytes	length	length of unicode string to follow
Variant	cell text	unicode string, not present if length = 0
4 bytes	horizontal alignment	horizontal alignment
4 bytes	vertical alignment	vertical alignment
1 byte	alpha	alpha color



Table 2–10: Slices resource block

Type	Name	Description
1 byte	red	red
1 byte	green	green
1 byte	blue	blue

## Photoshop 3.0 files

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

**Table 2–11: 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 'ic18' -16455 and 'ICN#' -16455 resources containing thumbnail images which will be shown in the Mac OS Finder.

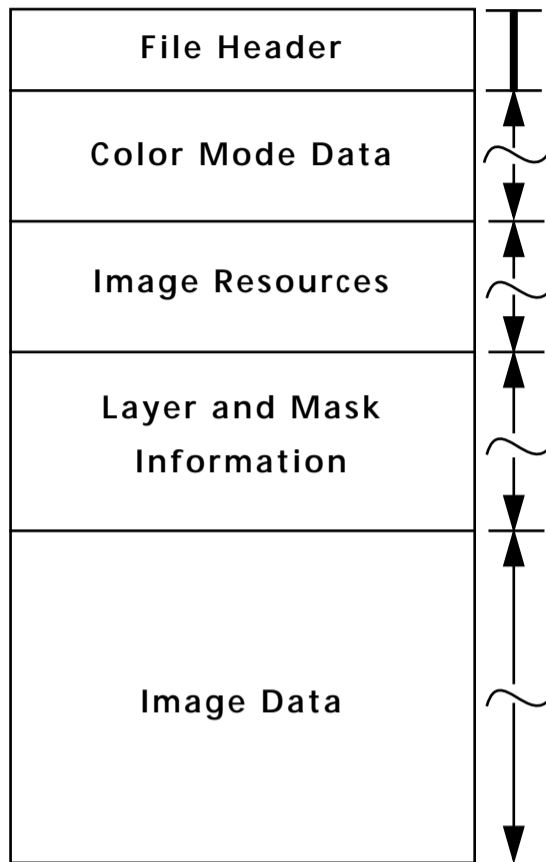
All of the data from Photoshop's File Info dialog is stored in 'ANPA' resource 10000. The data in this resource is stored as an IPTC-NAA record 2 and should be readable by various tools from Iron Mike. The DNPRV4.PDF document is found in the documentation folder of the SDK. For more information on the 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

<http://www.iptc.org>

## Photoshop 3.0 file format

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



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 2–12: 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 2–13: 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 2–14: 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 2–15: Layer and mask information**

Length	Name	Description
4 bytes	Length	Length of the miscellaneous information section.
Variable	Layers	Layer info. <a href="#">Layer info section on page 22</a>
Variable	Global layer mask	Global layer mask info. <a href="#">Global layer mask info on page 25</a>

### 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 2–16: Image data**

Length	Name	Description
2 bytes	Compression	Compression method. Raw data = 0, RLE compressed = 1, ZIP without prediction = 2, ZIP with prediction = 3.
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 2–15. The second part of this section contains layer mask data, which is described in table 2–22.

**Table 2–17: 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. <a href="#">Layer structure on page 22</a>
Variable	Pixel data	Channel image data for each channel in the order listed in the layers structure section. <a href="#">Channel image data on page 24</a>

**Table 2–18: 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. <a href="#">Layer records on page 22</a>

**Table 2–19: 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. <a href="#">Channel length info on page 23</a>
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 'smud' = exclusion 'div ' = color dodge 'idiv' = color burn
1 byte	Opacity	0 = transparent ... 255 = opaque

Table 2–19: Layer records (Continued)

Length	Name	Description
1 byte	Clipping	0 = base, 1 = non-base
1 byte	Flags	bit 0 = transparency protected bit 1 = visible bit 2 = obsolete bit 3 = 1 for Photoshop 5.0 and later, tells if bit 4 has useful information bit 4 = pixel data irrelevant to appearance of document
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.
40 bytes, 24 bytes, or 4 bytes if no layer mask.	Layer mask data	<u>Layer mask / adjustment layer data on page 23</u>
Variable	Layer blending ranges	<u>Layer blending ranges data on page 24</u>
Variable	Layer name	Pascal string, padded to a multiple of 4 bytes.
<i>These fields are new since version 4.0 of Adobe Photoshop: More new features where added for 5.0 and 6.0</i>		
Variable	Adjustment layer info	Series of tagged blocks containing various types of data. <u>Adjustment layer info on page 26</u> , <u>Effects Layer info on page 27</u> , <u>Type Tool Info on page 29</u> , <u>Unicode Layer name on page 30</u> , <u>Layer ID on page 30</u> , etc. Search this document for the four character code in the adjustment layer header.

Table 2–20: 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 2–21: Layer mask / adjustment layer data

Length	Name	Description
4 bytes	Size	Size of the data. This will be either 36, 20, or 0 (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	Only present if size = 20. Otherwise the following is present.
1 byte	Real Flags	Same as Flags information above.

Table 2–21: Layer mask / adjustment layer data

Length	Name	Description
1 byte	Real user mask background	0 or 255..
4 bytes	Top	Rectangle enclosing layer mask.
4 bytes	Left	
4 bytes	Bottom	
4 bytes	Right	

Table 2–22: 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
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 2–23: Channel image data

Length	Name	Description
2 bytes	Compression	0 = Raw Data, 1 = RLE compressed, ZIP without prediction = 2, ZIP with prediction = 3.
Variable	Image data	<p>If the compression code is 0, the image data is just the raw image data calculated as <math>((\text{LayerBottom} - \text{LayerTop}) * (\text{LayerRight} - \text{LayerLeft}))</math>. If the compression code is 1, the image data starts with the byte counts for all the scan lines in the channel <math>(\text{LayerBottom} - \text{LayerTop})</math>, 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.</p> <p>If the Layer's Size, and therefore the data, is odd, a pad byte will be inserted at the end of the row.</p> <p><i>New since version 4.0 of Adobe Photoshop:</i> If the layer is an adjustment layer, the channel data is undefined (probably all white.)</p>



Table 2–24: 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.
Variable	(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 [Image resource IDs on page 8](#)
2. URL ID. Image resource ID 1035 (0x040B). See [Image resource IDs on page 8](#).
3. Grid and guides information. See [Image resource IDs on page 8](#) and grid and guides resource format, earlier in this chapter.
4. Adjustment layers. See [Adjustment layer info on page 26](#).

**Table 2–25: Adjustment layer info**

Length	Name	Description
4 bytes	signature	always '8BIM'
4 bytes	key	OStype key for which adjustment type to use: 'lev1'=Levels 'curv'=Curves 'brit'=Brightness/contrast 'blnc'=Color balance 'hue '=Old Hue/saturation, Photoshop 4.0 'hue2'=New Hue/saturation, Photoshop 5.0 '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 the next chapter for information.

## Photoshop 5.0 file format

The Photoshop 5.0 file format is an extension of the Photoshop 4.0 and 3.0 file format. It is essentially the same, with some additional image resources and resource blocks. Additional resources for 5.0 are:

1. Thumbnail resource. Image resource ID 1036 (0x040C). See [Image resource IDs on page 8](#)
2. Global Angle resource. Image resource ID 1037 (0x040D). See [Image resource IDs on page 8](#).
3. Color Samplers resource. Image resource ID 1038 (0x040E). See [Image resource IDs on page 8](#), [Color Samplers resource block on page 13](#), and color samplers resource format, earlier in this chapter.
4. ICC Profile resource. Image resource ID 1039 (0x040F). See the ICC34.pdf and ICC34.h file for more information.
5. Watermark resource. Image resource ID 1040 (0x0410). See [Image resource IDs on page 8](#).
6. ICC UntaggedProfile resource. Image resource ID 1041 (0x0411). See [Image resource IDs on page 8](#).
7. Effects visible resource. Image resource ID 1042 (0x0412). See [Image resource IDs on page 8](#).
8. Spot halftone resource. Image resource ID 1043 (0x0413). See [Image resource IDs on page 8](#).
9. Document specific IDs seed number. Image resource ID 1044 (0x0414). See [Image resource IDs on page 8](#).
10. Unicode Alpha names resource. Image resource ID 1045 (0x0415). See [Unicode Layer name on page 30](#).

**Table 2–26: Effects Layer info**

Length	Name	Description
4 bytes	signature	always '8BIM'
4 bytes	key	always 'lrFX'
4 bytes	length	Length of effects data below rounded up to even 4 byte count.
2 bytes	version	always 0
2 bytes	effects count	Always 6. This includes the 6 effects in Photoshop 5.0. The next three items are repeated for each of the effects.
4 bytes	signature	always '8BIM'
4 bytes	effects signatures	OStype key for which adjustment type to use: 'cmnS'=common state 'dsdw'=drop shadow 'isdw'=inner shadow 'oglw'=outer glow 'iglw'=inner glow 'bevl'=bevel
Variable	data	See <a href="#">Effects layer, common state info on page 28</a> , <a href="#">Effects layer, drop shadow and inner shadow info on page 28</a> , <a href="#">Effects layer, outer and inner glow info on page 28</a> , and <a href="#">Effects layer, bevel info on page 28</a> .

Table 2–27: Effects layer, common state info

Length	Name	Description
4 bytes	size	always 7. size of next three items
4 bytes	version	always 0
1 byte	visible	always true
2 bytes	unused	always 0

Table 2–28: Effects layer, drop shadow and inner shadow info

Length	Name	Description
4 bytes	size	always 41. size of the remaining items
4 bytes	version	always 0
4 bytes	blur	Blur value in pixels.
4 bytes	intensity	Intensity as a percent
4 bytes	angle	Angle in degrees
4 bytes	distance	Distance in pixels
10 bytes	color	2 bytes for space followed by 4 * 2 byte color component
8 bytes	blend mode	4 bytes for signature and 4 bytes for the key
1 byte	enabled	effect enabled
1 byte	use global angle	Use this angle in all of the layer effects
1 byte	opacity	Opacity as a percent

Table 2–29: Effects layer, outer and inner glow info

Length	Name	Description
4 bytes	size	always 32. size of the remaining items
4 bytes	version	always 0
4 bytes	version	always 0
4 bytes	blur	Blur value in pixels.
4 bytes	intensity	Intensity as a percent
10 bytes	color	2 bytes for space followed by 4 * 2 byte color component
8 bytes	blend mode	4 bytes for signature and 4 bytes for the key
1 byte	enabled	effect enabled
1 byte	opacity	Opacity as a percent

Table 2–30: Effects layer, bevel info

Length	Name	Description
4 bytes	size	always 58. size of the remaining items
4 bytes	version	always 0
4 bytes	angle	Angle in degrees
4 bytes	strength	Depth in pixels
4 bytes	blur	Blur value in pixels.

Table 2–30: Effects layer, bevel info

Length	Name	Description
8 bytes	highlight blend mode	4 bytes for signature and 4 bytes for the key
8 bytes	shadow blend mode	4 bytes for signature and 4 bytes for the key
10 bytes	Highlight color	2 bytes for space followed by 4 * 2 byte color component
10 bytes	shadow color	2 bytes for space followed by 4 * 2 byte color component
1 byte	bevel style	
1 byte	highlight opacity	Highlight opacity as a percent
1 byte	shadow opacity	Shadow opacity as a percent
1 byte	enabled	effect enabled
1 byte	use global angle	Use this angle in all of the layer effects
1 byte	inverted	Up or down

Table 2–31: Type Tool Info

Length	Name	Description
4 bytes	signature	always '8BIM'
4 bytes	key	always 'tySh'
4 bytes	length	Length of Type tool info data below rounded up to even 4 byte count.
2 bytes	version	=1 for Photoshop 5.0
48 bytes	transform	6 * 8 double precision numbers for the transform information
Font information		
2 bytes	version	=6 for Photoshop 5.0
2 bytes	count	Count of faces, the next 8 fields are repeated for each count
2 bytes	mark	mark value
4 bytes	font type	font type data
Variable	name	pascal string of font name
Variable	family name	pascal string of font family name
Variable	style name	pascal string of font style name
2 bytes	script	script value
4 bytes	count	number of design axes vector to follow
4 bytes	design vector	design vector value
Style information		
2 bytes	count	number of styles to follow, the next 10 fields are repeated for each count
2 bytes	mark	mark value
2 bytes	face mark	face mark value
4 bytes	size	size value
4 bytes	tracking	tracking value

Table 2–31: Type Tool Info

Length	Name	Description
4 bytes	kerning	kerning value
4 bytes	leading	leading value
4 bytes	base shift	base shift value
1 byte	auto kern	auto kern on/off
1 byte	extra	only present in version <= 5
1 byte	rotate	rotate up/down
Text information		
2 bytes	type	type value
4 bytes	scaling factor	scaling factor value
4 bytes	char count	character count value
4 bytes	horizontal	horizontal placement
4 bytes	vertical	vertical placement
4 bytes	select start	select start value
4 bytes	select end	select end value
2 bytes	line count	number of items to follow, the next 5 fields are repeated for each line count
4 bytes	character count	character count value
2 bytes	orientation	orientation value
2 bytes	alignment	alignment value
2 bytes	double byte character	actual character as a double byte character
2 bytes	style	style value
2 bytes	color space	color space value
8 bytes	color data	4 * 2 byte color compenens
1 byte	anti alias	anti alias on/off

Table 2–32: Unicode Layer name

Length	Name	Description
4 bytes	signature	always '8BIM'
4 bytes	key	always 'luni'
4 bytes	length	length of the following fields
4 bytes	layer name length	length of the unicode string
Variable	unicode string	see <a href="#">Image resource IDs on page 8</a> for more information

Table 2–33: Layer ID

Length	Name	Description
4 bytes	signature	always '8BIM'
4 bytes	key	always 'lyid'
4 bytes	length	always 4
4 bytes	ID	see <a href="#">Image resource IDs on page 8</a> for more information

## Photoshop 6.0 file format

The Photoshop 6.0 file format is an extension of the Photoshop 5.0, 4.0 and 3.0 file format. It is essentially the same, with some additional image resources and resource blocks. Additional resources for 6.0 are:

1. Indexd Color Table. Image resource ID 1046 (0x0416). See [Image resource IDs on page 8](#)
2. Transparency Index. Image resource ID 1047 (0x0417). See [Image resource IDs on page 8](#).
3. Global Altitude. Image resource ID 1049 (0x0419).
4. Slices. Image resource ID 1050 (0x041A). See [Image resource IDs on page 8](#).
5. Workflow URL. Image resource ID 1051 (0x041B). See [Image resource IDs on page 8](#).
6. Jump to XPEP. Image resource ID 1052 (0x041C). See [Image resource IDs on page 8](#).
7. Alpha Identifier. Image resource ID 1053 (0x041D). See [Image resource IDs on page 8](#).
8. URL List. Image resource ID 1054 (0x041E). See [Image resource IDs on page 8](#).
9. Version Info. Image resource ID 1057 (0x0421). See [Image resource IDs on page 8](#).
10. Object based effects layer.
11. Patterns
12. Annotations

**Table 2–34: Object Based Effects Layer info**

Length	Name	Description
4 bytes	signature	always '8BIM'
4 bytes	key	always 'lfx2'
4 bytes	length	Length of effects data below rounded up to even 4 byte count.
4 bytes	object effects version	always 0
4 bytes	descriptor version	=16. For Photoshop 6.0.
Variable	Descriptor	See the description of descriptors in the Additional document chapter titled Actions

**Table 2–35: Patterns**

Length	Name	Description
4 bytes	signature	always '8BIM'
4 bytes	key	always 'Patt'
4 bytes	length	Length of the patterns information rounded up to even 4 byte count.

Table 2–35: Patterns

Length	Name	Description
Following is repeated for each pattern.		
4 bytes	length	length of this pattern
4 bytes	version	=1 for Photoshop 6.0
4 bytes	image 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.
4 bytes	point	vertical, 2 bytes and horizontal, 2 bytes
4 bytes	length	length of unicode string to follow
Variable	unicode string	name
Variable	unique ID	unique ID for this pattern
Variable	index color table	only present when image mode is indexed color
Variable	data	actual pattern data

Table 2–36: Annotations

Length	Name	Description
4 bytes	signature	always '8BIM'
4 bytes	key	always 'Anno'
4 bytes	length	Length of the annotation information rounded up to even 4 byte count.
2 bytes	major version	=2 for Photoshop 6.0
2 bytes	minor version	=1 for Photoshop 6.0
4 bytes	count	number of Annotations to follow
Following is repeated for each annotation		
4 bytes	length	length of this annotation
4 bytes	annotation type	'txtA' or 'sndA'. Either text or sound
1 byte	open	is the annotation open
1 byte	flags	flag information
2 bytes	optional blocks	=1 for Photoshop 6.0
16 bytes	icon rectangle	rectangle of icon location: top, left, bottom and right.
16 bytes	popup rectangle	rectangle of popup locations: top, left, bottom and right
10 bytes	color	2 bytes for space followed by 4 * 2 byte color component
Variable	author	Pascal string of author's name aligned to 2 bytes
Variable	name	Pascal string of name aligned to 2 bytes
Variable	mod Date	Pascal string of the mod Date aligned to 2 bytes
4 bytes	length	length of the following 3 fields including this field
4 bytes	key	'txtC' or 'sndM'. Either text or sound



Table 2–36: Annotations

Length	Name	Description
4 bytes	length	length of the next field
Variable	string	actual data for this annotation, the text is ASCII or uni-code string, the sound annotation is documented in the Acrobat File Format documentation in the Acrobat SDK and available at <a href="http://www.adobe.com">http://www.adobe.com</a>
Variable	padding	padding to align to multiple of 4 bytes

Table 2–37: Blend clipping elements

Length	Name	Description
4 bytes	signature	always '8BIM'
4 bytes	key	always 'c1b1'
4 bytes	length	Length of the information, =4 for Photoshop 6.0
1 byte	blend clipped elements	boolean
3 bytes	padding	

Table 2–38: Blend interior elements

Length	Name	Description
4 bytes	signature	always '8BIM'
4 bytes	key	always 'infx'
4 bytes	length	Length of the information, =4 for Photoshop 6.0
1 byte	blend interior elements	boolean
3 bytes	padding	

Table 2–39: Knockout setting

Length	Name	Description
4 bytes	signature	always '8BIM'
4 bytes	key	always 'knko'
4 bytes	length	Length of the information, =4 for Photoshop 6.0
1 byte	knockout	boolean
3 bytes	padding	

Table 2–40: Protected setting

Length	Name	Description
4 bytes	signature	always '8BIM'
4 bytes	key	always 'lspf'
4 bytes	length	Length of the information, =4 for Photoshop 6.0
4 bytes	protection	protection flags, bits 0 - 2 are used for Photoshop 6.0. Transparency, composite and position respectively.

Table 2–41: Sheet Color setting

Length	Name	Description
4 bytes	signature	always '8BIM'
4 bytes	key	always 'lclr'
4 bytes	length	Length of the information, =8 for Photoshop 6.0
4 * 2 bytes	color	Only the first color setting is used for Photoshop 6.0, rest are zeros

Table 2–42: Reference point

Length	Name	Description
4 bytes	signature	always '8BIM'
4 bytes	key	always 'fxrp'
4 bytes	length	Length of the information, =16 for Photoshop 6.0
2 * 8 bytes	point	2 double values for the reference point

Table 2–43: Gradient settings

Length	Name	Description
4 bytes	signature	always '8BIM'
4 bytes	key	always 'grdm'
4 bytes	length	Length of the information to follow
2 bytes	version	=1 for Photoshop 6.0
1 byte	reverse	is gradient reversed
1 byte	dither	is gradient dithered
4 bytes	name length	length of the unicode string to follow
Variable	name	name of the gradient as a unicode string, padded for even bytes
2 bytes	color stop count	number of color stops to follow
Following is repeated for each color stop		
4 bytes	location	location of color stop
4 bytes	midpoint	midpoint of color stop
2 bytes	color mode	mode for the color to follow
4 * 2 bytes	color	actual color for the stop
2 bytes	transparency stop count	number of transparency stops to follow
Following is repeated for each transparency stop		
4 bytes	location	location of transparency stop
4 bytes	midpoint	midpoint of transparency stop
2 bytes	opacity	opacity of transparency stop
2 bytes	expansion count	=2 for Photoshop 6.0
2 bytes	interpolation	interpolation if length above is non zero
2 bytes	length	=32 for Photoshop 6.0

Table 2–43: Gradient settings

Length	Name	Description
2 bytes	gradient mode	mode for this gradient
4 bytes	random seed	random number seed
2 bytes	show transparency	flag for showing transparency
2 bytes	use vector color	flag for using vector color
4 bytes	roughness	roughness factor
2 bytes	color model	color model
4 * 2 bytes	minimum color	minimum color values
4 * 2 bytes	maximum color	maximum color values
2 bytes	dummy	not used in Photoshop 6.0

Table 2–44: Section Divider setting

Length	Name	Description
4 bytes	signature	always '8BIM'
4 bytes	key	always 'lscT'
4 bytes	length	4 or 12
4 bytes	type	4 possible values, 0 = any other type of layer, 1 = open "folder", 2 = closed "folder", 3 = bounding section divider, hidden in the UI
Following is only present if length = 12		
4 bytes	signature	always '8BIM'
4 bytes	key	blending key option, <a href="#">Layer records on page 22</a>

Table 2–45: Channel Blending Restrictions setting

Length	Name	Description
4 bytes	signature	always '8BIM'
4 bytes	key	always 'brst'
4 bytes	length	length of data to follow
Following is repeated length / 4 times.		
4 bytes	channel number	channel number that is restricted

Table 2–46: Solid Color Sheet setting

Length	Name	Description
4 bytes	signature	always '8BIM'
4 bytes	key	always 'SoCo'

Table 2–46: Solid Color Sheet setting

Length	Name	Description
4 bytes	length	length of data to follow
4 bytes	version	=16. For Photoshop 6.0.
Variable	descriptor	Following is based on the Action file format structure documented in the Actions chapter. <a href="#">Descriptor structure on page 46</a>

Table 2–47: Pattern Fill setting

Length	Name	Description
4 bytes	signature	always '8BIM'
4 bytes	key	always 'PcFl'
4 bytes	length	length of data to follow
4 bytes	version	=16. For Photoshop 6.0.
Variable	descriptor	Following is based on the Action file format structure documented in the Actions chapter. <a href="#">Descriptor structure on page 46</a>

Table 2–48: Gradient Fill Setting

Length	Name	Description
4 bytes	signature	always '8BIM'
4 bytes	key	always 'GdFl'
4 bytes	length	length of data to follow
4 bytes	version	=16. For Photoshop 6.0.
Variable	descriptor	Following is based on the Action file format structure documented in the Actions chapter. <a href="#">Descriptor structure on page 46</a>

Table 2–49: Vector Mask setting

Length	Name	Description
4 bytes	signature	always '8BIM'
4 bytes	key	always 'vmsk'
4 bytes	length	length of data to follow
4 bytes	version	=3. For Photoshop 6.0.
4 bytes	flags	bit 1 = invert, bit 2 = not link, bit 3 = disable
The rest of the data is path components, loop until end of the length.		
Variable	paths	<a href="#">Path resource format on page 14</a>

Table 2–50: Type Tool Object setting

Length	Name	Description
4 bytes	signature	always '8BIM'
4 bytes	key	always 'TySh'
4 bytes	length	length of data to follow

Table 2–50: Type Tool Object setting

Length	Name	Description
2 bytes	version	=1. For Photoshop 6.0.
6 * 8 bytes	transform	xx, xy, yx, yy, tx, and ty respectively.
2 bytes	text descriptor version	=50. For Photoshop 6.0.
4 bytes	descriptor version	=16. For Photoshop 6.0.
Variable	descriptor	<u>Descriptor structure on page 46</u>
2 bytes	warp descriptor version	=1. For Photoshop 6.0.
4 bytes	descriptor version	=16. For Photoshop 6.0.
Variable	descriptor	<u>Descriptor structure on page 46</u>
4 * 8 bytes	warp rectangle as doubles	left, top, right, bottom respectively.

Table 2–51: Foreign effect ID

Length	Name	Description
4 bytes	signature	always '8BIM'
4 bytes	key	always 'ffxi'
4 bytes	length	length of data to follow
4 bytes	ID	ID of the Foreign effect.

Table 2–52: Layer Name Source setting

Length	Name	Description
4 bytes	signature	always '8BIM'
4 bytes	key	always 'lnsr'
4 bytes	length	length of data to follow
4 bytes	layer Name	ID for the layer name

Table 2–53: Pattern data

Length	Name	Description
4 bytes	signature	always '8BIM'
4 bytes	key	always 'shpa'
4 bytes	length	length of data to follow
4 bytes	version	=0. For Photoshop 6.0
4 bytes	count	Number of sets to follow
The following is repeated for the count above.		
4 bytes	signature	pattern signature
4 bytes	key	pattern key
4 bytes	pattern count	number of patterns in this set

Table 2–53: Pattern data

Length	Name	Description
1 byte	copy on sheet duplication	name says it all
3 bytes	padding	we like to make the file big
The following is repeated for the count of patterns above.		
4 bytes	color handling	prefer convert = 'conv', avoid conversion = 'avod', luminance only = 'lumi'
Variable	name	Pascal string name of the pattern
Variable	unicode name of the pattern	Unicode string name of the pattern
Variable	unique id	Pascal string of the unique identifier for the pattern

Table 2–54: Meta Data setting

Length	Name	Description
4 bytes	signature	always '8BIM'
4 bytes	key	always 'shmd'
4 bytes	length	length of data to follow
4 bytes	count	count of meta data items to follow
The following is repeated for the count above.		
4 bytes	signature	signature of the data
4 bytes	key	key of the data
1 byte	copy on sheet duplication	name says it all
3 bytes	padding	we like to make the file big
4 bytes	length	length of data to follow
Variable	raw data	undocumented data

Table 2–55: Layer Data

Length	Name	Description
4 bytes	signature	always '8BIM'
4 bytes	key	always 'Layer'
Variable	Layer Info structure	<u><a href="#">Layer info section on page 22</a></u>

## 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 2–56: 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 2–57: 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. Photoshop 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. $(\text{columns} * \text{depth} + 7) / 8$ =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 2–58: 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 rearranged 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). Photoshop 5.0 stores the ICC Profile in TIFF files under tag number 34675 (see the ICC34.pdf file from the International Color Consortium located in the documentation section).

Photoshop 6.0 uses 4 more tags that are not defined in the TIFF v6 specification. They are: tSubIFD (330), tJPEGTables (347), tImageData (37724), and tAnnotations (50255). tSubIFD is documented in the TIFF-PM6.pdf file as a PageMaker extension. tAnnotations are discussed earlier in this chapter, the TIFF tag does not include the first 12 bytes of the Annotation. The details for the remaining tags are in the tables at the end of this section.

tImageData is a null-terminated string identifying the type of data in the tag followed by the data for the tag. See the tagged resources found earlier in this document.

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 Uchisaiwai-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 2-35 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 'ic18' -16455 and 'ICN#' -16455 resources containing thumbnail images which will be shown in the Mac OS Finder.

Photoshop compression 1 = uncompressed, 2 = CCITT, 5 = LZW, 7 = JPEG, 8 = ZIP, and 32773 = Pack Bits.

Table 2-59: 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, 7, 8, 32773	1, 5, 7, 8
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

# 3. Additional File Formats

Besides documents that the user creates in Adobe Photoshop (discussed in the previous chapter), there are a number of additional files used by Photoshop to store information about colors, brushes, etc. These can be saved to files and loaded into Photoshop for use at a later time or with different images.

Each file has a unique file type and file extension associated with it. Photoshop for Macintosh will recognize either, but does not require the use of the extension. Photoshop for Windows will look for the given file extension automatically; this can be overridden.

Many of the files, but not all, have version numbers written as short integers in the first two bytes of the file.

Under the Mac OS, all information is stored in the data forks of Photoshop's load files. The files are completely interchangeable with Windows or any other platform.



**Note:** Consistent byte ordering is required across platforms when reading and writing load files. Photoshop stores multi-byte values with the high-order bytes first, big-endian, like on 680x0 systems with the Mac OS. This is opposite of the way it is done on Intel systems with Windows. For more information, see “Macintosh and Windows development” in chapter 2 of *Photoshop SDK Guide.pdf*.

## Actions

Action files have been around since version 4.0 of Photoshop. This description will only cover the output produced by Photoshop 6.0. The object effects for Photoshop 6.0 use the actions mechanism to output information to the PSD file format.

**Table 2-60: Action file types**

OS	Filetype/extension
Mac OS	8BAC
Windows	.ATN

**Table 2-61: Action file format**

Length	Name	Description
4 bytes	version	=16. For Photoshop 6.0.
4 bytes	length	length of unicode string to follow
Variable	set name	unicode string of the actions set name.
1 byte	expanded	is set expanded for the actions palette
4 bytes	set children	number of actions in action set
<i>Following is repeated for each action set</i>		
2 bytes	index	index of action set
1 byte	shift key	shift key needed for keyboard shortcut
1 byte	command key	command key needed for keyboard shortcut
2 bytes	color index	color information
4 bytes	length	length of unicode string to follow
Variable	action name	unicode string of the action name
1 byte	expanded	is action expanded in the actions palette
4 bytes	children	number of items in action
<i>Following is repeated for each action</i>		
1 byte	expanded	is action expanded in the actions palette
1 byte	enabled	action is enabled
1 byte	with dialog	dialogs should be displayed
1 byte	dialog options	options for displaying dialogs
4 bytes	identifier	'TEXT' or 'long'
Variable	event	4 bytes of length followed by the string or the 4 bytes of itemID if identifier is 'long'
Variable	event for Display	length of the string to follow or zero if identifier is 'long'
Variable	dictionary name	4 bytes of length followed by the string
4 bytes	has descriptor	-1 for a descriptor to follow or 0 for none.
Variable	Descriptor	see Descriptor structure table for details

Table 2-62: Descriptor structure

Length	Name	Description
4 bytes	length	length of unicode string to follow
Variable	classID	classID as a unicode string
Variable	classID2	4 bytes for the length of the string to follow or if zero the next item will be a 4 byte classID
4 bytes	number of items	items in descriptor
<i>Following is repeated for each item in descriptor</i>		
Variable	key	4 bytes for the length of the string to follow or if zero the next item will be a 4 byte key
4 bytes	type	OSType key for type to use: 'obj ' =Reference 'Objc' =Descriptor 'VLLs' =List 'doub' =Double 'UntF' =Unif float 'TEXT' =String 'Enmr' =Enumerated 'long' =Integer 'bool' =Boolean 'GlbO' =GlobalObject same as Desriptor 'Cls' =Class 'GlbC' =GlobalClass 'alis' =Alias
Variable	item Type	see the tables below for each possible type

Table 2-63: Reference Structure

Length	Name	Description
4 bytes	length	number of items
<i>Following is repeated for each item in reference</i>		
4 bytes	type	OSType key for type to use: 'prop'=Property 'Cls'=Class 'Enmr'=Enumerated 'rele'=Offset 'Idnt'= Identifier 'indx'=Index 'name'=Name
Variable	item Type	see the tables below for each possible Reference type

Table 2-64: Property Structure

Length	Name	Description
Variable	classID as string	4 bytes for the length of the unicode string to follow or if zero the next item will be a 4 byte classID
Variable	classID	4 bytes for the length of the string to follow or if zero the next item will be a 4 byte classID
Variable	keyID	4 bytes for the length of the string to follow or if zero the next item will be a 4 byte property key

Table 2–65: Unit float structure

Lenght	Name	Description
4 bytes	units	units the following value is in
8 bytes	double	actual value

Table 2–66: Double structure

Lenght	Name	Description
8 bytes	double	actual value

Table 2–67: Class structure

Lenght	Name	Description
Variable	classID as string	4 bytes for the length of the unicode string to follow or if zero the next item will be a 4 byte classID
Variable	classID	4 bytes for the length of the string to follow or if zero the next item will be a 4 byte classID

Table 2–68: String structure

Lenght	Name	Description
4 bytes	length	length of the unicode string to follow
Variable	unicode string	string value as unicode string

Table 2–69: Enumerated structure

Lenght	Name	Description
Variable	classID as string	4 bytes for the length of the unicode string to follow or if zero the next item will be a 4 byte classID
Variable	keyID	4 bytes for the length of the string to follow or if zero the next item will be a 4 byte key
Variable	typeID	4 bytes for the length of the string to follow or if zero the next item will be a 4 byte property type
Variable	enum	4 bytes for the length of the string to follow or if zero the next item will be a 4 byte property enum

Table 2–70: Offset structure

Lenght	Name	Description
Variable	classID as string	4 bytes for the length of the unicode string to follow or if zero the next item will be a 4 byte classID
Variable	classID	4 bytes for the length of the string to follow or if zero the next item will be a 4 byte class
4 bytes	offset	value of the offset

Table 2–71: Boolean structure

Lenght	Name	Description
1 bytes	value	you guessed it

Table 2–72: Alias structure

Lenght	Name	Description
4 bytes	length	length of data to follow
Variant	handle	data in the Handle, FSSpec for Macintosh and a handle to a string to the full path on Windows

Table 2–73: List structure

Lenght	Name	Description
4 bytes	items	number of items in the list
<i>Following is repeated for each item in list</i>		
4 bytes	type	OSType key for type to use: 'obj ' =Reference 'Objc' =Descriptor 'VlLs' =List 'doub' =Double 'UntF' =Unif float 'TEXT' =String 'Enmr' =Enumerated 'long' =Integer 'bool' =Boolean 'GlbO' =GlobalObject same as Desriptor 'Clss' =Class 'GlbC' =GlobalClass 'alis' =Alias
Variable	item Type	see the tables below for each possible type



## Arbitrary Map

Arbitrary Map files are loaded and saved in Photoshop's "Curves" dialog.

**Table 3-1: Arbitrary map file types**

OS	Filetype/extension
Mac OS	8BLT
Windows	.AMP

There is no version number written in the file, and the file must be an even multiple of 256 bytes long. Each 256 bytes is a lookup table, where:

1. The first byte of the table corresponds to byte zero of the image.
2. The last byte of the table corresponds to byte 255 of the image.
3. A `NULL` table that has no effect on an image is a linear table of bytes from 0 to 255.

If the file has one table, it is applied to the image's channels according to these priorities:

1. If the image has a master composite channel, the table is applied to it. If not, then:
2. If the image has a single active channel, the table is applied to it. If not, then:
3. If the image has no composite channel and more than one active channel, the table is not applied.

If the file has exactly three tables, it is applied to the image's channels according to these priorities:

1. The tables are assumed to represent RGB lookups. They are applied to the first three channels in the image, leaving the master composite untouched. Or:
2. If the image has a single active channel, the tables are converted to grayscale and the result is applied to the active channel. Or:
3. The first table is treated as a master. The remaining tables are applied to the image channels in turn (second table is applied to first channel, third table is applied to second channel, etc.).

### Single active channels

Photoshop handles single active channels in a special fashion. When saving a map applied to a single channel, only one table is written to the file. Similarly, when reading a file for application to a single active channel, the master table is the one that will be used on that channel. This allows easy application of a single file to both composite and grayscale images.

## Brushes

Brushes settings files are loaded and saved in Photoshop's "Brushes" palette. These are typically stored in the `Goodies/Brushes & Patterns` sub-folder in the Mac OS, or the `Brushes` sub-directory in Windows.

**Table 3-2: Brushes file types**

OS	Filetype/extension
Mac OS	8BBR
Windows	.ABR

**Table 3-3: Brushes file format**

Length	Name	Description
2 bytes	version	=1. Short integer.
2 bytes	count	A short integer indicating how many brushes are in the remainder of the file.
Variable	brushes	Two types of brushes are currently supported: elliptical, computed brushes and sampled brushes. Computed brushes are created with the "New Brush" command; sampled brushes are created from selected image data using the "Define Brush" command. See table 3-4.

**Table 3-4: Brush components**

Length	Name	Description
2 bytes	type	A short integer indicating the type of brush. 1=Computed brush; 2=Sampled brush. Other values are currently undefined.
4 bytes	size	A long integer indicating the number of bytes in the remainder of the brush definition. Photoshop uses this information to skip over brush types that it doesn't understand.
<i>size</i> bytes	data	The contents depend on the type of brush. Computed brush data is always 14 bytes; sampled brush data varies in size depending on the image data that makes up the brush tip.

**Table 3-5: Computed brush parameters**

Length	Name	Description
4 bytes	miscellaneous	Long integer. Ignored.
2 bytes	spacing	Short integer from 0...999 where 0=no spacing.
2 bytes	diameter	Short integer from 1...999.
2 bytes	roundness	Short integer from 0...100.
2 bytes	angle	Short integer from -180...180.
2 bytes	hardness	Short integer from 0...100.

**Table 3-6: Sampled brush parameters**

Length	Name	Description
4 bytes	miscellaneous	Long. Ignored.
2 bytes	spacing	Short integer from 0...999 where 0=no spacing.
1 byte	anti-aliasing	0=no anti-aliasing when applied; 1=anti-alias when applied. Brushes with sampled data taller or wider than 32 pixels will never be anti-aliased.

Table 3-6: Sampled brush parameters (Continued)

Length	Name	Description
8 bytes	bounds	Rectangle: Four short integers giving the bounds of the sampled data in the order <code>top</code> , <code>left</code> , <code>bottom</code> , <code>right</code> .
16 bytes	bounds-long	Rectangle, same as <code>Bounds</code> , but in four long integers.
2 bytes	depth	Depth of the sample data. Always 8.
Variable	image data	If the bounds are taller than 16384 pixels, the data is broken into 16384-line chunks. Each chunk is streamed as shown in table 3-7.

Table 3-7: Sampled brush image data structure

Length	Name	Description
2 bytes	compression	0=Raw data, 1=RLE compressed.
Variable	data	<p>The brush tip image data is a single plane of grayscale data, stored in scanline order, with no pad bytes.</p> <p>If <code>compression=0</code>, the data is just the raw image data.</p> <p>If <code>compression=1</code>, the data starts with the byte counts for all the scan lines (equal to the number of rows, as described by the <code>bounds</code>), 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 <code>PackBits</code>, and the TIFF standard.</p>

## CMYK Setup

CMYK settings files are loaded and saved in Photoshop 5.0's "CMYK Setup" dialog, via the "Preferences" sub-menu. Photoshop 5.0 combines the Ink Color Setup file and the Separation Setup file from Photoshop 4.0

**Table 3-8: CMYK file types**

OS	Filetype/extension
Mac OS	8BIC
Windows	.API

**Table 3-9: CMYK setup file format**

Length	Name	Description
2 bytes	version	=7. Short integer.
27*2 bytes	ink colors	Nine sets of three short integers specifying the $xY$ (CIE) values for the inks and their combinations. The inks are specified in the order cyan, magenta, yellow, magenta–yellow (red), cyan–yellow (green), cyan–magenta (blue), cyan–magenta–yellow, followed by the white and black points. Each set is written in the order $xY$ where:  $x=0...10000$ , representing $0.0...1.0000$ . $y=1...10000$ , representing $0.0001...1.0000$ . $z=0...20000$ , representing $0.00...200.00$ .
2 bytes	dot gain	Short integer from -10...40, representing -10%...40%.
1 byte	use curves	=1 for curves table present.
1 byte	(filler)	(zero)
13*4*2 bytes	Curves value	Only present if "use curves" = 1. 4 sets of 13 short integers specifying the cyan, magenta, yellow, and black curve percentages from the Dot Gain Curves dialog. $0...1000$ , representing $0.0...100.0$ %
Variable	Separation Setup	See table 3-55

## Color Table

Color Table files are loaded and saved in Photoshop's "Color Table" dialog (used with Indexed Color images), and can also be loaded into the "Colors" palette.

**Table 3-10: Color table file types**

OS	Filetype/extension
Mac OS	8BCT
Windows	.ACT

There is no version number written in the file. The file is exactly 768 bytes long, and contains 256 RGB colors:

1. The first color in the table is index zero.
2. There are three bytes per color in the order Red, Green, Blue.

If loaded into the "Colors" palette, the colors will be installed in the color swatch list as RGB colors.

## Colors

Colors files are loaded and saved in Photoshop's "Colors" palette. These are typically stored in the `Palettes` sub-directory in Windows.

**Table 3-11: Colors file types**

OS	Filetype/extension
Mac OS	8BCO
Windows	.ACO

**Table 3-12: Colors file format**

Length	Name	Description
2 bytes	version	=1. Short integer.
2 bytes	count	Short integer indicating how many colors are in the file.
count * 10 bytes	colors	Each color is 10 bytes composed of a <code>space</code> byte and <code>data</code> . Described in table 3-13.

**Table 3-13: Color structure**

Length	Name	Description
2 bytes	space	A short integer indicated the color space the color belongs to as shown in table 3-14.
8 bytes	data	Four short unsigned integers with the actual color data. if the color does not require four values to specify, the extra values are undefined and should be written as zeros. See table 3-14.

**Table 3-14: Color space IDs**

Color ID	Name	Description
0	RGB	The first three values in the color data are <i>red</i> , <i>green</i> , and <i>blue</i> . They are full unsigned 16-bit values as in Apple's <code>RGBColor</code> data structure. Pure red=65535,0,0.
1	HSB	The first three values in the color data are <i>hue</i> , <i>saturation</i> , and <i>brightness</i> . They are full unsigned 16-bit values as in Apple's <code>HSVColor</code> data structure. Pure red=0,65535, 65535.
2	CMYK	The four values in the color data are <i>cyan</i> , <i>magenta</i> , <i>yellow</i> , and <i>black</i> . They are full unsigned 16-bit values. 0=100% ink. Pure cyan=0,65535,65535,65535.
7	Lab	The first three values in the color data are <i>lightness</i> , <i>a chrominance</i> , and <i>b chrominance</i> .  Lightness is a 16-bit value from 0...10000. The chrominance components are each 16-bit values from -12800...12700. Gray values are represented by chrominance components of 0. Pure white=10000,0,0.
8	grayscale	The first value in the color data is the gray value, from 0...10000.

Photoshop allows the specification of custom colors, such as those colors that are defined in a set of custom inks provided by a printing ink manufacturer. These colors can be stored in the "Colors" palette and streamed to and from load files. The details of a custom color's color data fields are not public and should be treated as a black box.

Table 3-15 gives the color space IDs currently defined by Photoshop for some custom color spaces.

**Table 3-15: Custom color spaces**

Color ID	Name
3	Pantone matching system
4	Focoltone colour system
5	Trumatch color
6	Toyo 88 colorfinder 1050
10	HKS colors (European Photoshop only)

## Command Settings File

Commands settings files are loaded and saved in Photoshop 3.0's "Commands" palette. This feature supplants the Function Key feature of Photoshop 2.5. The Commands palette buttons are simple mappings to Photoshop menu items, with optional function key shortcut and colorization.



**Note:** The Commands palette does not exist in Photoshop 4.0. Its functionality has been absorbed into the "Actions" palette. This section is provided for backwards compatibility and reference only.

**Table 3-16: Command settings file types**

OS	Filetype/extension
Mac OS	8BFK
Windows	.ACM

**Table 3-17: Command settings file format**

Length	Name	Description
2 bytes	version	=2. Short integer.
2 bytes	count	Number of command records in the file. There are no pad bytes between records.
Variable	records	Command records, one after the other. Described in table 3-18.

**Table 3-18: Command record structure**

Length	Name	Description
4 bytes	command ID	Command ID. Must be zero. Obsolete.
2 bytes	function key ID	Integer from -15...15. Positive numbers map directly onto the numbered function keys (F1, F2, etc.). Negative numbers indicate that the shift key must be used for the shortcut (Shift-F1, Shift-F2, etc.). Zero means the button has no keyboard shortcut. On Windows systems, values outside of -12 to 12 will be ignored as standard Windows systems have 12 function keys on the keyboard. Windows systems will also map 1 to 0, as the F1 key is reserved for Help. These numbers should be unique across all entries in a Commands file. Photoshop will ignore duplicates.
2 bytes	color index	Each command button can be assigned a color with which its background will be tinted when drawn. There are eight predefined colors: 0=None; 1=Red; 2=Orange; 3=Yellow; 4=Green; 5=Blue; 6=Purple; 7=Gray.



Table 3-18: Command record structure

Length	Name	Description
1 byte	title matching flag	<p>Boolean flag indicating button title updating off/on. For example, a button assigned to the “Layers” palette would change text from “Show Layers” to “Hide Layers” automatically as the state of the palette and the actual menu item changes.</p> <p>0=Don’t update. Button title has been changed from the menu item text by the user and shouldn’t change unless by user.</p> <p>1=Update. Button title should automatically be updated to match the command’s current menu item text.</p>
Variable	button title	Pascal-style string, with no pad bytes. This is the title of the button that will be drawn on the Command palette. Usually matches menu item text.
Variable	command key	<p>Pascal-style string, with no pad bytes. This is the key for finding the menu item in Photoshop’s menus. To distinguish menu items which could be duplicated on different menus, a key may include the title of the menu itself followed by a colon (“Mode:RGB Color”). This text is displayed in the options dialog for the button, but not on the Commands palette itself. Even if <code>TitleMatching=1</code>, the button text never contains the menu title qualifier.</p>

## Curves

Curves settings files are loaded and saved in Photoshop's "Curves" dialog and "Black Generation" curve dialog (from within Separation Setup Preferences). Curves files can also be loaded into any of Photoshop's transfer function dialogs, such as the Duotone Curve dialog from within Duotone Options.



**Note:** When loaded into a transfer function dialog, only the first curve in a Curves file is used.

Table 3-19: Curves file types

OS	Filetype/extension
Mac OS	8BSC
Windows	.CRV

Table 3-20: Curves file format

Length	Name	Description
2 bytes	version	=1. Short integer.
2 bytes	count	Short integer indicating how many curves are in the file.
Variable	curves	Curves data, one after the other. Described in table 3-21.

Table 3-21: Curves data structure

Length	Name	Description
2 bytes	point count	Short integer from 2...19 indicating how many points are in the curve.
point count * 4 bytes	curve points	Each curve point is a pair of short integers where the first number is the output value (vertical coordinate on the Curves dialog graph) and the second is the input value. All coordinates have range 0 to 255. See also <i>Null curves</i> , below.

### Null curves

A NULL curve (no change to image data) is represented by the following five-number, ten-byte sequence in a file:

```
2 0 0 255 255.
```

### Displaying ink percentages

Photoshop allows the option of displaying ink percentages instead of pixel values; this is a display option only and the internal data is unchanged, with 100% ink equal to image data of 0 and 0% ink equal to image data of 255.

### Curves data order

1. The first curve is a master curve that applies to all the composite channels (RGB) when in composite image mode.
2. The remaining curves apply to the active channels in order: curve two applies to channel one, curve three applies to channel two, etc., up until curve 17, which applies to channel 16.

### Indexed color

The exception to the normal order, and the reason there are up to 19 curves, is when the mode is Indexed color. In this case:

1. The first curve is a master curve.
2. The next three curves are created for the Red, Green, and Blue portions of the image's color table, and they are applied to the first channel.
3. The remaining curves apply to any remaining alpha channel that is active: for instance, if channel two is active, curve five applies to it; if channel three is active, curve six applies to it, etc., up until curve 19, which applies to channel 16.

### Single active channels

Photoshop handles single active channels in a special fashion. When saving the curves applied to a single channel, the settings are stored into the master curve, at the beginning of the file. Similarly, when reading a curves file for application to a single active channel, the master curve is the one that will be used on that channel. This allows easy application of a single file to both RGB and grayscale images.

### Photoshop 3.0 and later Curves files and Photoshop 2.0

Photoshop 3.0 and later can write Curves files that Photoshop 2.0 will not be able to read, because version 3.0 and later of Photoshop's active channel support is different from Photoshop 2.0's. There could be more active channels in a Curves dialog than 2.0 supported.

For maximum compatibility with version 2.0, Photoshop 3.0 and later will pad the file with `NULL` curves to always write at least five curves. The presence of extraneous `NULL` curves will not affect a load operation.

### Photoshop 3.0 and later Curves files and Photoshop 2.5

It is possible, however rare, to create a Curves load file with Photoshop 3.0 or later that cannot be read by Photoshop 2.5. Version 3.0 and later of Photoshop allows a maximum of 24 channels per document, Photoshop 2.5 allows 16.

## Custom kernel

Kernel settings files are loaded and saved in Photoshop's "Custom filter" dialog.

**Table 3-22: Custom kernel file types**

OS	Filetype/extension
Mac OS	8BCK
Windows	.ACF

### Format:

1. There is no version number written in the file.
2. The file is expected to be exactly 54 bytes long, representing 27 short integers, described in table 3-23.

**Table 3-23: Custom filter structure**

Length	Name	Description
50 bytes	weights	The first 25 values are the custom weights from $-999\dots999$ , applied to pixels offset from each pixel by $[-2,-2]$ to $[2,2]$ . The values progress through horizontal offsets first. See <i>Weight offset progression</i> below.
27*2 bytes	ink colors	Nine sets of three short integers specifying the $xyY$ (CIE) values for the inks and their combinations. The inks are specified in the order cyan, magenta, yellow, magenta–yellow (red), cyan–yellow (green), cyan–magenta (blue), cyan–magenta–yellow, followed by the white and black points. Each set is written in the order $xyY$ where:  $x=0\dots10000$ , representing $0.0\dots1.0000$ . $y=1\dots10000$ , representing $0.0001\dots1.0000$ . $Y=0\dots20000$ , representing $0.00\dots200.00$ .
2 bytes	scale	Short integer from $1\dots9999$ .
2 bytes	offset	Short integer from $-9999\dots9999$ .

### Weight offset progression

This is sample matrix for the weight offset progression:

```
{[-2,-2],[-1,-2],[ 0,-2],[ 1,-2],[ 2,-2],
 [-2,-1],[-1,-1],[ 0,-1],[ 1,-1],[ 2,-1],
 [-2, 0],[-1, 0],[ 0, 0],[ 1, 0],[ 2, 0],
 [-2, 1],[-1, 1],[ 0, 1],[ 1, 1],[ 2, 1],
 [-2, 2],[-1, 2],[ 0, 2],[ 1, 2],[ 2, 2]}
```

## Duotone options

Duotone settings files are loaded and saved in Photoshop’s “Duotone Options” dialog.

**Table 3-24: Duotone file types**

OS	Filetype/extension
Mac OS	8BDT
Windows	.ADO

**Table 3-25: Duotone file format**

Length	Name	Description
2 bytes	version	=1. Short integer.
2 bytes	count	Short integer from 1...4 indicating how many plates are in the duotone spec. 1=Monotone; 2=Duotone; 3=Tritone; 4=Quadtone.
4*10 bytes	ink colors	Four ink colors, regardless of the number of plates. The contents of the colors beyond the last plate specified by Count are undefined. Each color is 10 bytes and described in table 3-26. It is identical to the format in a Colors load file.
4*64 bytes	ink names	Four ink names, regardless of the number of plates. Each name is streamed as a Pascal-style string with a length byte followed by the string name. Names may not be more than 63 characters. Each name is padded to occupy 64 bytes, including the length byte. Any names beyond the last plate specified by Count should be empty, size=0.
4*28 bytes	ink curves	Four ink curves, regardless of the number of plates. Described in table 3-28.
2 bytes	dot gain	=20. Short integer. Kept for compatability with Photoshop 2.0. Ignored.
11*10 bytes	overprint colors	Eleven ink colors, regardless of the number of plates. The number of defined overprints depends on Count.  Monotones=no overprint colors. Duotones=one overprint color. Tritones=four overprint colors. Quadtones=11 overprint colors. The contents of the colors beyond the last defined overprint are undefined. Each color is 10 bytes and described in table 3-26. It is identical to the format in a Colors load file.

**Table 3-26: Duotone color structure**

Length	Name	Description
2 bytes	space	A short integer indicated the color space the color belongs to as shown in table 3-27.
8 bytes	data	Four short unsigned integers with the actual color data. if the color does not require four values to specify, the extra values are undefined and should be written as zeros. See table 3-27.

Table 3-27: Duotone color space IDs

Color ID	Name	Description
0	RGB	The first three values in the color data are <i>red</i> , <i>green</i> , and <i>blue</i> . They are full unsigned 16-bit values as in Apple's <code>RGBColor</code> data structure. Pure red=65535,0,0.
1	HSB	The first three values in the color data are <i>hue</i> , <i>saturation</i> , and <i>brightness</i> . They are full unsigned 16-bit values as in Apple's <code>HSVColor</code> data structure. Pure red=0,65535, 65535.
2	CMYK	The four values in the color data are <i>cyan</i> , <i>magenta</i> , <i>yellow</i> , and <i>black</i> . They are full unsigned 16-bit values. 0=100% ink. Pure cyan=0,65535,65535,65535.
7	Lab	The first three values in the color data are <i>lightness</i> , <i>a chrominance</i> , and <i>b chrominance</i> .  Lightness is a 16-bit value from 0...10000. The chrominance components are each 16-bit values from -12800...12700. Gray values are represented by chrominance components of 0. Pure white=10000,0,0.
8	grayscale	The first value in the color data is the gray value, from 0...10000.

Table 3-28: Ink curves structure

Length	Name	Description
26 bytes	transfer curve	Array of 13 short integers from 0...1000 representing 0.0...100.0. All but the first and last value may be -1, representing no point on the curve. See <i>Null transfer curve</i> below.
2 bytes	override	=0. Short integer for compatibility. Ignored by Photoshop 3.0.

**Null transfer curve**

Any curves beyond the last plate specified by `Count` should be equal to the NULL curve. A NULL transfer curve looks like this:

```
0, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1000.
```

## Halftone screens

Halftone Screens settings files are loaded and saved in Photoshop's Halftone Screens dialog (from within Page Setup).

**Table 3-29: Halftone screen file types**

OS	Filetype/extension
Mac OS	8BHS
Windows	.AHS

**Table 3-30: Halftone screens file format**

Length	Name	Description
2 bytes	version	=5. Short integer.
4*18 bytes	screens	Four screen descriptions. See table 3-31.
variable (see description)	curves	For every screen which has a custom spot function, the text of the PostScript function is written here. The functions are written one after the other with no header information, in the same order as the screen settings. The size of each custom spot is the absolute value of its negative shape code.

**Table 3-31: Halftone screen parameter structure**

Length	Name	Description
4 bytes	frequency value	Ink's screen frequency, in lines per inch. Binary fixed point value with 16 bits representing the integer and fractional parts from 1.0...999.999.
2 bytes	frequency scale	Units for the screen frequency. Lines per inch = 1; lines per centimeter = 2. Only affects display, not <code>Frequency Value</code> .
4 bytes	angle	Angle for screen. Binary fixed point value with 16 bits representing the integer and fractional parts from -180.0000 ... 180.0000, measured in degrees.
2 bytes	shape code	Code representing the shape of the halftone dots. 0=Round; 1=Ellipse; 2=Line; 3=Square; 4=Cross; 6=Diamond. Custom shapes are represented by a negative number. The absolute value of the negative number is the size in bytes of the custom spot function described in table 3-30.
4 bytes	miscellaneous	=0. Not currently used by Photoshop.
1 byte	accurate screens	Boolean. 1=Use accurate screens; 0=Use other.
1 byte	default screens	Boolean. 1=Use printer's default screens; 0=Use other.

## Hue/Saturation

Hue/Saturation settings files are loaded and saved in Photoshop's "Hue/Saturation" dialog. The Hue/Saturation file format has been updated for Photoshop 5.0 and later. See table 3-35 below for the updated information.

**Table 3-32: Hue/saturation file types**

OS	Filetype/extension
Mac OS	8BHA
Windows	.HSS

**Table 3-33: Hue/saturation file format**

Length	Name	Description
2 bytes	version	=2 for Photoshop 5.0 and later. =1 for Photoshop 4.0 and earlier. Short integer.
1 byte	mode	Boolean. 0=Use settings for hue-adjustment; 1=Use settings for colorization.
1 byte	padding	Padding byte must be present but is ignored by Photoshop.
6 bytes	colorization	<p>Photoshop 5.0</p> <p>The actual values are stored for the new version. Hue is -180...180, Saturation is 0...100, and Lightness is -100...100.</p> <p>Photoshop 4.0</p> <p>Three short integers Hue, Saturation, and Lightness from -100...100. The user interface represents hue as -180...180, saturation as 0...100, and Lightness as -100...1000, as the traditional HSB color wheel, with red=0.</p>
Variable	hue-saturation settings	<p>90 bytes. Photoshop 5.0 see table 3-35.</p> <p>Photoshop 4.0 see table 3-34.</p> <p>42 bytes. Three sets of seven short integers, from -100...100. Described in table 3-34.</p>

**Table 3-34: Hue-saturation settings, Photoshop 4.0**

Length	Name	Description
14 bytes	hue settings	<p>Seven short integers. The first value is the master hue change, followed by six values.</p> <p>For RGB and CMYK, those values apply to each of the six hexants in the HSB color wheel: those image pixels nearest to red, yellow, green, cyan, blue, or magenta. These numbers appear in the user interface from -60...60, however the slider will reflect each of the possible 201 values from -100...100.</p> <p>For Lab, the first four of the six values are applied to image pixels in the four Lab color quadrants, yellow, green, blue, and magenta. The other two values are ignored (=0). The values appear in the user interface from -90 to 90.</p>
14 bytes	saturation settings	Seven short integers. The first is a master saturation value. The other six are applied to pixels exactly the same as the hue settings.
14 bytes	lightness settings	Seven short integers. The first is a master lightness value. The other six are applied to pixels exactly the same as the hue settings.



Table 3-35: Hue-saturation settings. Photoshop 5.0

Length	Name	Description
6 bytes	master values	Master hue, saturation and lightness values.
6 sets of: 4 range values followed by 3 settings values		
8 bytes	range values	<p>For RGB and CMYK, those values apply to each of the six hexants in the HSB color wheel: those image pixels nearest to red, yellow, green, cyan, blue, or magenta. These numbers appear in the user interface from -60...60, however the slider will reflect each of the possible 201 values from -100...100.</p> <p>For Lab, the first four of the six values are applied to image pixels in the four Lab color quadrants, yellow, green, blue, and magenta. The other two values are ignored (=0). The values appear in the user interface from -90 to 90.</p>
6 bytes	settings values	

## Ink colors setup

Ink Colors settings files are loaded and saved in Photoshop 3.0's "Ink Colors Setup" dialog, via the "Preferences" sub-menu. Photoshop 5.0 combines the Ink Color Setup file and the Separation Setup file into one file with the same extension and a version number of 7. Please refer to the CMYK setup section for more information.

**Table 3-36: Ink colors file types**

OS	Filetype/extension
Mac OS	8BIC
Windows	.API

**Table 3-37: Ink colors setup file format**

Length	Name	Description
2 bytes	version	=4. Short integer.
27*2 bytes	ink colors	Nine sets of three short integers specifying the $xyY$ (CIE) values for the inks and their combinations. The inks are specified in the order cyan, magenta, yellow, magenta–yellow (red), cyan–yellow (green), cyan–magenta (blue), cyan–magenta–yellow, followed by the white and black points. Each set is written in the order $xyY$ where:  $x=0...10000$ , representing $0.0...1.0000$ . $y=1...10000$ , representing $0.0001...1.0000$ . $Y=0...20000$ , representing $0.00...200.00$ .
4*2 bytes	gray balance	Four short integers from 50...200, representing 0.5 to 2.00. Specifies the gray color balance for CMYK.
2 bytes	dot gain	Short integer from -10...40, representing -10%...40%.

## Levels

Levels settings files are loaded and saved in Photoshop's "Levels" dialog.

**Table 3-38: Levels file types**

OS	Filetype/extension
Mac OS	8BLS
Windows	.ALV

**Table 3-39: Levels file format**

Length	Name	Description
2 bytes	version	=2. Short integer.
290 bytes	levels records	29 sets of levels, each level contains 10 bytes of five short integers. Described in table 3-40.

**Table 3-40: Level record structure**

Length	Name	Description
2 bytes	input floor	Short integer from 0...253.
2 bytes	input ceiling	Short integer from 2...255.
2 bytes	output floor	Short integer from 0...255. Matched to <code>input floor</code> .
2 bytes	output ceiling	Short integer from 0...255.
2 bytes	gamma	Short integer from 10...999 representing 0.1...9.99. Applied to all image data.

### Level record sets order

1. The first set of levels is the master set that applies to all of the composite channels (RGB) when in composite image mode.
2. The remaining sets apply to the active channels individually; set two applies to channel one, the set three to channel two, etc., up until set 25, which applies to channel 24.
3. Sets 28 and 29 are reserved and should be set to zeros.

### Indexed color

The exception to the normal order is when the mode is Indexed:

1. The first set is a master set.
2. The next three sets are created for the Red, Green, and Blue portions of the image's color table, and they are applied to the first channel.
3. The remaining sets apply to any remaining alpha channels that are active: for instance, if channel two is active, set five applies to it; if channel three is active, set six applies to it, etc., up until channel 27, which applies to channel 24.
4. Sets 28 and 29 are reserved and should be set to zeros.

### Single active channels

Photoshop handles single active channels in a special fashion. When saving the levels applied to a single channel, the settings are stored into the master set, at the beginning of the file. Similarly, when reading a levels file for application to a single active channel, the master levels are the ones that will be used on that channel. This allows easy application of a single file to both RGB and grayscale images.

### **Photoshop 3.0 and later Levels files and Photoshop 2.5**

There are two versions of the Levels file format. Photoshop 3.0 and later reads both but only writes version 2. Because the maximum number of channels was increased in Photoshop 3.0 from 16 to 24, Photoshop 3.0 and later actually writes a longer Levels file than Photoshop 2.5. Photoshop 2.5 is still capable of reading these files and ignores the extra data.

## Monitor setup

Monitor settings files are loaded and saved in Photoshop's "Monitor Setup" dialog, via the "Preferences" sub-menu in Photoshop 3.0, and under the File menu in Photoshop 4.0.

**Table 3-41: Monitor setup file types**

OS	Filetype/extension
Mac OS	8BMS
Windows	.AMS

**Table 3-42: Monitor setup file format**

Length	Name	Description
2 bytes	version	=2. Short integer.
2 bytes	gamma	Short integer from 75...300 representing 0.75...3.00.
2*2 bytes	white point	Two short integers as CIE chromaticity coordinates: $x, y$ . $x=0...10000$ representing 0.0...1.0000. $y=1...10000$ representing 0.0001...1.0000.
6*2 bytes	phosphors	Three sets of two integers giving $x, y$ coordinates of the red, green, and blue phosphors. $x=0...10000$ representing 0.0...1.0000. $y=1...10000$ representing 0.0001...1.0000. In the order <i>red</i> $x$ , <i>red</i> $y$ ; <i>green</i> $x$ , <i>green</i> $y$ ; <i>blue</i> $x$ , <i>blue</i> $y$ .

## Replace color/Color range

Replace Color settings files are loaded and saved in Photoshop's "Replace Color" dialog. They are also used to load and save settings from the "Color Range" dialog.

**Table 3-43: Replace color/Color range file types**

OS	Filetype/extension
Mac OS	8BXT
Windows	.AXT

**Table 3-44: Replace color/Color range file format**

Length	Name	Description
2 bytes	version	=1. Short integer.
2 bytes	color space	Short integer indicating what space the color components are in. 7=Lab color, 8=grayscale. No other values are supported.
6 bytes	component ranges	Six unsigned byte values representing the range of colors within which a pixel's color must fall to be considered selected for color replacement, or color range selecting. Described in table 3-45.
2 bytes	fuzziness	Short integer from 0...200 controlling how colors close to selected colors are affected.
6 bytes	transform settings	When used with Replace Color: Three short integers from -100...100. Described in table 3-46. When used with Color Range: Writes zeros into the three short integers and ignores.

**Table 3-45: Component range structure**

Length	Name	Description
1 byte	low endpoint 1	if Lab (color space=7): low endpoint of L value if grayscale (color space=8): low endpoint of gray range
1 byte	high endpoint 1	if Lab: high endpoint of L value if grayscale: 0
1 byte	low endpoint 2	if Lab: low endpoint of a chrominance value if grayscale: 0
1 byte	high endpoint 2	if Lab: high endpoint of a chrominance value if grayscale: 0
1 byte	low endpoint 3	if Lab: low endpoint of b chrominance value if grayscale: low endpoint of gray range
1 byte	high endpoint 3	if Lab: high endpoint of b chrominance value if grayscale: high endpoint of gray range

**Table 3-46: Replace color transform settings**

Length	Name	Description
2 bytes	hue	Short integer from -100...100. Hue change.
2 bytes	saturation	Short integer from -100...100. Saturation change.
2 bytes	lightness	Short integer from -100...100. Lightness change.

## Scratch Area

Scratch Area settings files are loaded and saved in Photoshop's Scratch palette.

**Table 3-47: Scratch area file types**

OS	Filetype/extension
Mac OS	8BSR
Windows	.ASR

**Table 3-48: Scratch area file format**

Length	Name	Description
2 bytes	version	=1. Short integer.
Variable	data	Scratch area data in the form of RGB image data. The three planes are written one after the other in the order <i>red</i> , <i>green</i> , <i>blue</i> . Described in table 3-49.

**Table 3-49: Scratch area data structure**

Length	Name	Description
16 bytes	bounds	Four long integers giving the bounds of the scratch data rectangle in the order <i>top</i> , <i>left</i> , <i>bottom</i> , <i>right</i> . Photoshop 3.0 has a fixed Scratch palette size and this will always be [0, 0, 89, 200]
2 bytes	depth	Depth of the current data plane. Always 8.
Variable	image data	Image data. Described in table 3-50.

**Table 3-50: Scratch area image data structure**

Length	Name	Description
2 bytes	compression	0=Raw data, 1=RLE compressed.
Variable	data	Each plane of the image data is stored in scanline order, with no pad bytes. If <code>compression=0</code> , the data = raw data. If <code>compression=1</code> , the data starts with the byte counts for all the scan lines (equal to the number of rows, as described by the bounds), 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 <code>PackBits</code> , and the TIFF standard.

## Selective color

Selective Color settings files are loaded and saved in Photoshop's "Selective Color" dialog.

**Table 3-51: Selective color file types**

OS	Filetype/extension
Mac OS	8BSV
Windows	.ASV

**Table 3-52: Selective color file format**

Length	Name	Description
2 bytes	version	=1. Short integer.
2 bytes	correction method	Short integer. 0=Apply color correction in relative mode; 1=Apply color correction in absolute mode.
80 bytes	plate corrections	Ten eight-byte correction records, described in table 3-53.

**Table 3-53: Plate correction structure**

Length	Name	Description
2 bytes	cyan	Short integer from -100...100. Amount of cyan correction.
2 bytes	magenta	Short integer from -100...100. Amount of magenta correction.
2 bytes	yellow	Short integer from -100...100. Amount of yellow correction.
2 bytes	black	Short integer from -100...100. Amount of black correction.

### Record order

1. The first record is ignored by Photoshop 3.0 and is reserved for future use. It should be set to all zeroes.
- 2...10. The rest of the records apply to specific areas of colors or lightness values in the image, in the following order: reds, yellows, greens, cyans, blues, magentas, whites, neutrals, blacks.



## Separation setup

Separation settings files are loaded and saved in Photoshop's "Separation Setup" dialog, via the "Preferences" sub-menu. Photoshop 5.0 combines the Ink Color Setup file and the Separation Setup file into one file with the same extension and a version number of 7. Please refer to the CMYK setup section for more information.

**Table 3-54: Separation file types**

OS	Filetype/extension
Mac OS	8BSS
Windows	.ASP

**Table 3-55: Separation file format**

Length	Name	Description
2 bytes	version	=300. Short integer.
2 bytes	separation type	Boolean. 0=UCR separations; 1=GCR separations.
2 bytes	black limit	Short integer from 0...100 giving the black ink limit.
2 bytes	total limit	Short integer from 200...400 giving the total ink limit.
2 bytes	UCA amount	Short integer from 0...100 giving the undercolor addition for GCR separations.
Variable	black generation curve	Spline curve detailed in table 3-56. Identical to the Curves data format in table 3-21.

**Table 3-56: Black generation curve data structure**

Length	Name	Description
2 bytes	point count	Short integer from 2...19 indicating how many points are in the curve.
2* point count bytes	curve points	Each curve point is a pair of short integers where the first number is the output value (vertical coordinate on the Black Generation dialog graph) and the second is the input value. All coordinates have range 0 to 255. See <i>Null curves</i> .

### Null curves

A NULL curve (no change to image data) is represented by the following five-number, ten-byte sequence in a file:

```
2 0 0 255 255.
```



#### Note:

The black generation curve and the UCA limit must both be present even if the separation type is set to UCR (=0).

## Separation tables

Separation Table files are loaded and saved in Photoshop's "Separation Tables" dialog.

**Table 3-57: Separation tables file types**

OS	Filetype/extension
Mac OS	8BST
Windows	.AST

### Format:

1. If the size of the file is  $33 * 33 * 33 * 4$ , then the file consists only of an Lab→CMYK table as currently documented.
2. If the size of the file is  $(33 * 33 * 33 + 256) * 3$ , then the file consists only of a CMYK→Lab table as currently documented.
3. Otherwise, the file has the format listed in table 3-58.

**Table 3-58: Separation table file format**

Length	Name	Description
2 bytes	version	=300. Short integer.
1 byte	has Lab to CMYK	Boolean. 0=No; 1=Contains Lab→CMYK table.
1 byte	has CMYK to Lab	Boolean. 0=No; 1=Contains CMYK→Lab table.
$33*33*33*4$ bytes	Lab to CMYK table	If <code>hasLabtoCMYK=1</code> then this section contains CMYK colors for $33*33*33$ Lab colors. The CMYK colors are written in interleaved order, one byte each ink. 0=100%, 255=0%. See <i>Generating Lab source colors</i> below.
$(33*33*33 + 256)*3$ bytes	CMYK to Lab table	If <code>hasCMYKtoLab=1</code> then this section contains Lab colors for $33*33*33+256$ CMYK colors. The Lab colors are written in interleaved order, one byte per component. See <i>Generating CMYK source colors</i> below.
1 byte	has gamut table	Boolean. 0=No; 1=gamut table follows.
1 byte	filler	If <code>hasGamutTable=0</code> then this byte will not be present. If <code>hasGamutTable=1</code> then this byte should be set to 1 for compatibility.
$((33*33*33L)+7)>>3$ bytes	gamut table	If <code>hasGamutTable=0</code> then this field will not be present. If <code>hasGamutTable=1</code> then this is the gamut table. The gamut table is a bit table indexed in the same way as the Lab→CMYK table with the high bit of the first byte at index 0. See <i>Testing for bits in the gamut table</i> , below.

**Generating Lab source colors**

The Lab colors that are the source colors can be generated from the Lab->CMYK table with the following routine:

```
for (i = 0; i < 33; i++)
  for (j = 0; j < 33; j++)
    for (n = 0; n < 33; n++)
      {
        L = Min (i * 8, 255);
        a = Min (j * 8, 255);
        b = Min (n * 8, 255);
      }
```

**Generating CMYK source colors**

The CMYK colors that are the source colors can be generated from the CMYK->Lab table with the following routine:

```
for (i = 0; i < 33; i++)
  for (j = 0; j < 33; j++)
    for (n = 0; n < 33; n++)
      {
        c = Min (i * 8, 255);
        m = Min (j * 8, 255);
        y = Min (n * 8, 255);
        k = 255;
      }
```

```
for (i = 0; i < 256; i++)
  {
    c = 255;
    m = 255;
    y = 255;
    k = i;
  }
```

**Testing for bits in the gamut table**

To test the bit at `bitIndex`, use table:

```
([bitIndex >> 3] & (0x0080 >> (bitIndex & 0x07))) != 0.
```

`bitIndex` itself is calculated in the same way you would calculate an index into the Lab->CMYK table.

A result of 1 indicates that the color is in gamut and 0 indicates that it is out of gamut.

## Transfer function

Transfer Function settings files are loaded and saved in Photoshop's "Duotone Curve" dialog from within "Duotone Options" and "Transfer Function" dialogs from within Page Setup. Transfer Function files can also be loaded into any of Photoshop's curves dialogs, such as the Curves color adjustment dialog.

**Table 3-59: Transfer function file types**

OS	Filetype/extension
Mac OS	8BTF
Windows	.ATF

**Table 3-60: Transfer function file format**

Length	Name	Description
2 bytes	version	=4. Short integer.
112 bytes	functions	There are four transfer functions in the file, described in table 3-61.

**Table 3-61: Transfer function structure**

Length	Name	Description
26 bytes	curve	Array of 13 short integers from 0...1000 representing 0.0...100.0. All but the first and last value may be -1, representing no point on the curve. See <i>Null transfer curve</i> below.
2 bytes	override	Boolean. 0=Let printer supply curve; 1=Override printer's default transfer curve.

### Null transfer curve

Any curves beyond the last plate specified by `Count` should be equal to the NULL curve. A NULL transfer curve looks like this:

```
0, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1000.
```



**Note:** The file always contains four functions. When writing the printer transfer functions for grayscale images, for instance, Photoshop writes four copies of the single transfer function specified in the user interface.



## Symbols

.ABR 50  
.ACF 60  
.ACM 56  
.ACO 54  
.ACT 52  
.ADO 60  
.AHS 63  
.ALV 67  
.AMP 49  
.AMS 69  
.API 66  
.ASP 73  
.ASR 71  
.AST 74  
.ASV 72  
.ATF 76  
.ATN 77  
.AXT 70  
.CRV 58  
.HSS 64

## Numerics

8BAC 77  
8BBR 50  
8BCK 60  
8BCO 54  
8BCT 52  
8BDT 60  
8BFK 56  
8BHA 64  
8BHS 63  
8BIC 66  
8BIM 8  
8BLS 67  
8BLT 49  
8BMS 69  
8BPS 19  
8BSC 58  
8BSR 71  
8BSS 73  
8BST 74  
8BSV 72  
8BTF 76  
8BXT 70

## A

accurate screens 63  
Actions 45  
Adjustment layer 23  
Adobe Premiere 40  
Alias 46, 48  
alis 46, 48  
angle 50, 63  
Anno 32  
ANPA 18, 43  
anti-aliasing 50

## B

Background color 8  
bevel 27

- bevl 27
- binary/ascii 39
- BitsPerSample 43
- bitspixel 12
- black 72
- black generation curve 73
- black limit 73
- Blend mode key 22
- Blend mode signature 22
- blnc 26
- block size 39
- bool 46, 48
- Boolean 46, 48
- Border information 8
- bounds 51, 71
- bounds-long 51
- Brightness/contrast 26
- brit 26
- brs 35
- brushes 50
- button title 57

## C

- Channel destination range 24
- Channel ID 23
- Channel length info 22
- Channel source range 24
- Channels 19
- Class 46, 48
- clbl 33
- Clipping 23
- Clipping path 10
- cmnS 27
- CMYK 54, 62
- CMYK to Lab table 74
- CMYK->Lab 75
- Color 9
- Color balance 26
- Color components 25
- Color data 20
- color index 56
- color space 70
- Color transfer functions 8
- colorization 64
- ColorMap 43
- colors 54
- columns 19, 39
- command ID 56
- command key 57
- common state 27
- component ranges 70
- Composite gray blend destination 24
- Composite gray blend source 24
- compressedsize 12
- compression 21, 24, 43, 51, 71
- Copyright 9
- Copyright flag 9
- correction method 72
- count 50, 54, 56, 58, 61
- curv 26
- curve 76
- curve points 58, 73

curves 26, 58, 63  
cyan 72

## D

data 50  
default screens 63  
depth 19, 39, 51, 71  
Descriptor 46, 48  
Desriptor 46, 48  
diameter 50  
DisplayInfo 8  
Document 9  
dot gain 61, 66  
DotRange 43  
doub 46, 48  
Double 46, 48  
dsdw 27, 46, 48  
Duotone 8  
Duotone transfer functions 8

## E

Effects 9  
encapsulated PostScript files 26  
Enmr 46, 48  
Enumerated 46, 48  
EPS 9  
Extra data size 23  
ExtraSamples 43

## F

fDirection 11  
ffxi 37  
fGridCycle 11  
fGuideCount 11  
filler 74  
FillOrder 43  
fLocation 11  
Focoltone 55  
format 12  
framesPerSec 40  
frequency scale 63  
frequency value 63  
function key ID 56  
functions 76  
fuzziness 70  
fVersion 11  
fxrp 34

## G

gamma 67, 69  
gamut table 74  
GAP SDK tech notes mailing list 6  
GdFI 36  
GetProperty() 14  
GlbC 46, 48  
GlbO 46, 48  
Global 9  
GlobalClass 46, 48  
GlobalObject 46, 48  
gray balance 52, 66  
grayscale 8, 54, 62



grdm 34  
Grid and guides information 9

## H

halftoning 8  
hardness 50  
has CMYK to Lab 74  
has gamut table 74  
has Lab to CMYK 74  
height 12, 40  
high endpoint 70  
HKS colors 55  
HSB 54, 62  
hue 26, 70  
hue settings 64  
Hue/saturation 26  
hue2 26  
hue-saturation settings 64

## I

ICC 9  
icl8 18, 43  
ICN# 18, 43  
Identifier 46  
Idnt 46  
IFD 43  
iglw 27  
image data 51, 71  
ImageDescription 43  
ImageLength 43  
ImageWidth 43  
Index 46  
indx 46  
infx 33  
ink colors 52, 60, 61, 66  
ink curves 61  
ink names 61  
InkSet 43  
inner glow 27  
inner shadow 27  
input ceiling 67  
input floor 67  
Integer 46, 48  
Invert 26  
invt 26  
IPTC-NAA 9  
IPTC-NAA 42  
isdw 27, 46, 48

## J

JFIF 12  
JPEG 9

## K

Kind 25  
knko 33

## L

Lab 54, 62  
Lab to CMYK table 74

- Lab→CMYK 75
- Layer 22
- Layer blending ranges 23
- Layer mask data 23
- Layer name 23
- Layer top 22
- Layers 20
- Layers structure 22
- Layr 38
- lclr 34
- leading 40
- Length 20
- Levels 26
- levels records 67
- levl 26
- lfx2 31, 36
- lightness 70
- lightness settings 64
- List 46, 48
- Insr 37, 38
- load files
  - Actions 77
  - Arbitrary Map 49
  - Brushes 50
  - CMYK setup 66
  - Color Table 52
  - Colors 54
  - Command Settings File 56
  - Curves 58
  - Custom kernel 60
  - Duotone options 60
  - Halftone screens 63
  - Hue/Saturation 64
  - Levels 67
  - Monitor setup 69
  - Replace color/Color range 70
  - Scratch Area 71
  - Selective color 72
  - Separation setup 73
  - Separation tables 74
  - Transfer function 76
- load files, description 44
- long 46, 48
- low endpoint 70
- lsct 35
- lspf 33

## M

- magenta 72
- Masks 20
- mode 19, 39, 64

## N

- Name 46
- name 46
- NewSubFileType 43
- numFrames 40

## O

- obj 46, 48
- Offset 46

- offset 60
- oglw 27, 46, 48
- One 9
- Opacity 22, 25
- outer glow 27
- output ceiling 67
- output floor 67
- Overlay color space 25
- overprint colors 61
- override 62, 76

## P

- packing 40
- pad channels 39
- padding 64
- Pantone 55
- Path 10
- Path Information 10
- Patt 31
- phosphors 69
- PhotometricInterpretation 43
- Photoshop EPS files 26
- PICT 18, 42
- Pixel data 22
- PlanarConfiguration 43
- planes 12
- plate corrections 72
- point count 58, 73
- post 26
- Posturize 26
- Predictor 43
- Premiere 40
- Print 10
- Print flags 8, 10
- PtFl 36

## Q

- Quick Mask 9

## R

- Reference 46, 48
- rele 46
- ResolutionInfo 8
- ResolutionUnit 43
- Resources 20
- RGB 54, 62
- roundness 50
- rows 19, 39
- RowsPerStrip 43

## S

- SamplesPerPixel 43
- saturation 70
- saturation settings 64
- scale 60
- screens 63
- selc 26
- Selective color 26
- separation type 73
- shape code 63
- shmd 38

shpa 37  
Signature 19  
size 12, 50  
SoCo 35  
space 54, 61  
spacing 50  
Spot 9  
STR 43  
STR# 18, 42  
String 46, 48  
StripByteCounts 43  
StripOffsets 43

## T

TEXT 18, 43, 46, 48  
Threshold 26  
thrs 26  
Thumbnail 9  
Thumbnail resource 9  
TileByteCounts 43  
TileLength 43  
TileOffsets 43  
TileWidth 43  
title matching flag 57  
total limit 73  
Toyo 88 colorfinder 55  
transfer curve 62  
transform settings 70  
Trumatch 55  
type 50  
TySh 36  
tySh 29

## U

UCA amount 73  
Unicode 9  
Unif float 46, 48  
UntF 46, 48  
URL 9

## V

version 19, 45, 50, 52, 54, 56, 58, 61, 63, 64, 66, 67, 69, 70, 72, 73, 74, 76  
VILs 46, 48  
vmsk 36

## W

weights 60  
white point 69  
width 12, 40  
widthbytes 12

## X

XResolution 43

## Y

yellow 72  
YResolution 43