



## **ICC profile tagging specifications**

### **HELIOS developer specification**

13-05-98

Number of pages: 13

(including extracts from Adobe's and the ICC's original documentation about profile tagging)

---

---

# ICC profile tagging specifications



This documentation describes the ICC profile tagging support in HELIOS EtherShare OPI 2.0.

## General remarks

HELIOS OPI 2.0 supports tagging of ICC profiles for various image formats. The image files can be tagged either by value (the profile is embedded into the image) or by reference (the file name of the profile is stored in the image).

The ICC defines tagging by value only for TIFF and PICT files. Tagging by value for EPSF files has been defined by Adobe. These three tagging options are described here too, but you should also refer to the documentation provided by the ICC, Apple, and Adobe.

HELIOS ICC profile references are defined as:

14 ASCII Byte Date/Time record immediately followed by the file name:

`yyyymmddhhmmss + filename`

The Date/Time record is taken from the ICC profile header during tagging time. A Date/Time record consisting only of ASCII 0 is taken as wildcard date and time. The file name may be specified in UNIX, Macintosh, or PC syntax. OpenImage Filename conventions may also be used.

An example:

The ICC profile reference is

`19951122092310ICC-Profiles:Printer:HELIOS:Euro 2.5 UCR-370`

Date/Time record is

`19951122092310` which translates to: 22. November 1995 09h23m10s

Filename is given in Macintosh syntax as:

`ICC-Profiles:Printer:HELIOS:Euro 2.5 UCR-370`

Embedded ICC profiles are stored “as is” starting with their header size element. The size of the profile is stored in the first four bytes in MSB first byte order and the value includes the size element (see also the documentation submitted by the ICC).

For some image formats the structure of ICC profile tag data is described using the C programming language. The following data types are used:

```
uint8      an unsigned 8 bit byte
uint16     an unsigned 16 bit integer
uint32     an unsigned 32 bit integer
AsciiInt   char[12] holding the ASCII representation of an integer
           12 digits in size with leading zeros.
```

All integer values are stored MSB first.

## Tagging for JPEG files

ICC profile tagging by reference for the JPEG image format is done by using a HELIOS private marker. As soon as the ICC and other vendors define a common format for storing ICC profile information in JPEG image files, this format will be adapted by HELIOS.

The HELIOS private marker will be found in the JPEG header as:

```
struct HeliosJPEGResource {
    uint8    type[4];           // See below: resource types
    uint8    creator[4];       // Always 0x48 0x4f 0x50 0x49 ("HOPI")
    uint16   len;              // Length of following data
    uint8    data[1];         // Data follow immediately
}

struct HeliosJPEGMarker {
    uint8    markerID[2];     // always 0xFF 0xEB
    uint16   markerLength;    // overall length of marker as defined
                                // for the JPEG format
    char     magic[7];        // always "HELIOS" with terminating zero byte
    uint16   version;         // always 0x0065 (version 101)
    uint8    type;            // Should be 0x01 (oiObjImage)
    uint8    quality;         // Should be 0x03 (oiObjQual)
    HeliosJPEGResource rsrcs[1]; // Resources follow immediately.
                                // The number of resources can be
                                // calculated from the marker length.
}
```

Resource type:

```
0x48 0x53 0x49 0x52 ("HSIR") for HELIOS ICC profile reference.
```

Example (from Hexdump):

```
00000410:    ff eb 00 55 48 45 4c 49 4f 53 00 00 65 01    ...UHELIOS..e.
00000420: 03 48 53 49 52 48 4f 50 49 00 3a 31 39 39 36 30    .HSIRHOPI.:19960
00000430: 36 32 37 31 31 31 35 31 31 49 43 43 2d 50 72 6f    627111511ICC-Pro
00000440: 66 69 6c 65 73 3a 50 72 69 6e 74 65 72 3a 48 45    files:Printer:HE
00000450: 4c 49 4f 53 3a 45 75 72 6f 20 32 2e 36 20 55 43    LIOS:Euro 2.6 UC
00000460: 52 2d 33 37 30 00 00 00 00                                R-370....
```

## Tagging for Scitex CT files

ICC profile tagging by reference or value for Scitex CT image files has been defined by Scitex and HELIOS.

```
struct HeliosCTResource {
    uint8      type[4];      // See below: resource types
    uint8      creator[4];   // Always 0x48 0x4f 0x50 0x49 ("HOPI")
    AsciiInt   offset;      // Offset to real data relative to the
                          // start of the control block
    AsciiInt   length;      // Length of real data which is appended
                          // at the end of the image.
}
```

Resource types:

0x48 0x53 0x49 0x52 ("HSIR") for HELIOS ICC profile reference data.

and

0x48 0x53 0x49 0x56 ("HSIV") for Embedded ICC profile

The resources will be placed in the last 768 Bytes of the parameter block beginning with offset 256. Up to 24 resources can be stored.

Please also refer to the Scitex CT HandShake format description which is available from Scitex.

Example (from Hexdump):

```
00000500: 48 53 49 52 48 4f 50 49 2b 30 30 30 30 31 32 33  HSIRHOPI+0000123
00000510: 30 38 34 38 2b 30 30 30 30 30 30 30 30 30 36 30  0848+00000000060
00000520: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....

...

0012c800: 31 39 39 36 30 37 30 35 31 30 35 33 32 30 49 43  19960705105320IC
0012c810: 43 2d 50 72 6f 66 69 6c 65 73 3a 50 72 69 6e 74  C-Profiles:Print
0012c820: 65 72 3a 48 45 4c 49 4f 53 3a 4a 61 70 61 6e 20  er:HELIOS:Japan
0012c830: 32 2e 36 20 55 43 52 2d 33 37 30 00              2.6 UCR-370.
```

## Tagging for PICT files

ICC profile tagging by value for Apple PICT image files is handled as follows:

Embedded profiles will be included into PICT images with one or more LongComment-Opcodes (0x00a1) depending on the size of the profile.

A selector is used to describe the type of the data:

- 0: Beginning of a V2.0 Profile; data follow...
- 1: Continuation of a V2.0 Profile; data follow...
- 2: End of a V2.0 Profile; no data follow.

Please refer to the Apple QuickDraw documentation for information on the structures used in PICT files.

Example 1: Embedding a 20k profile:

```
PicComment: 0x00a1, // 2 Byte; LongComment
Kind: 224, // 2 Byte; V2.0 Profile
DataSize: 20k + 4, // 2 Byte; 20k data + 4 Byte selector
Selector: 0, // 4 Byte; Beginning of a V2.0 Profile
Data ... // 20k Profile Data
PicComment: 0x00a1, // 2 Byte; LongComment
Kind: 224, // 2 Byte; V2.0 Profile
DataSize: 4, // 2 Byte; 4 Byte selector
Selector: 2 // 4 Byte; End of a V2.0 Profile
```

Example 2: Embedding a 50k profile:

```
PicComment: 0x00a1, // 2 Byte; LongComment
Kind: 224, // 2 Byte; V2.0 Profile
DataSize: 32k + 4, // 2 Byte; 32k (32k - 4 data + 4 Byte selector)
Selector: 0, // 4 Byte; Beginning of a V2.0 Profile
Data ... // 32k - 4 ProfileData
PicComment: 0x00a1, // 2 Byte; LongComment
Kind: 224, // 2 Byte; V2.0 Profile
DataSize: 18k + 8, // 2 Byte; 18k + 4 Byte data + 4 Byte selector
Selector: 1, // 4 Byte; Continuation of a V2.0 Profile
Data ... // 18k + 4 ProfileData
PicComment: 0x00a1, // 2 Byte; LongComment
Kind: 224, // 2 Byte; V2.0 Profile
DataSize: 4, // 2 Byte; 4 Byte selector
Selector: 2 // 4 Byte; End of a V2.0 Profile
```

## Tagging for Adobe Photoshop files

ICC profile tagging by reference for Adobe Photoshop 2.5, 3.0, and 4.0 image files is done by using an own resource in the Photoshop resource block. As soon as Adobe defines a common format for storing ICC profile information in Photoshop native image files, this format will be adapted by HELIOS.

The HELIOS ICC profile reference resource is added to the resource block of the image using the following type and creator signatures:

```
Type: "HSIR"
Creator: "HOPI"
```

The format of the data is:

```
For "HSIR": HELIOS ICC profile Reference
```

Please refer to the Adobe Photoshop SDK documentation for information on the structures used in Adobe Photoshop formats.

## Tagging for EPSF files

ICC profile tagging by reference for EPSF image formats is done by using a private DSC comment. As soon as the ICC or Adobe define a common format for storing ICC profile reference information in EPSF image files, this format will be adapted by HELIOS.

Tagging by reference is realized by the comment “%HELICCIImageProfileRef” placed in the comment section.

Syntax:

```
%HELICCIImageProfileRef: yyyyymmddhhmmss (filename)
```

Example:

```
%HELICCIImageProfileRef: 19951016095500 (ICC-Profiles:Scanner:CCIR-EBU-RGB)
```

Embedded profiles are placed immediately after the comment section using the well-defined Adobe ICC profile related DSC comments:

Syntax:

```
%%BeginICCPProfile: (<id>) <n> Hex Lines  
% <hex-line>  
% ...  
%%EndICCPProfile
```

Example:

```
%%BeginICCPProfile: (ICC-Profiles:Scanner:CCIR-EBU-RGB) 9524 Hex Lines  
% 0004a6806170706c0200000073636e72524742204c61622007cb000b00100013  
...  
% e0e1e2e3e4e5e6e7e8e9eaebeceedeef0f1f2f3f4f5f6f7f8f9fafbfcfdfeff  
%%EndICCPProfile
```

Please refer to the Adobe EPSF ICC documentation for further information on embedding ICC profiles in EPSF.

## Tagging for TIFF files

ICC profile tagging by reference for TIFF image files is done by using a private TIFF tag. As soon as the ICC or Adobe define a common format for storing ICC profile reference information in TIFF image files, this format will be adapted by HELIOS.

```
Tag number: 34841 (HELIOS assigned tag)  
Tag type: ASCII
```

Format of the data is:

```
HELIOS ICC profile reference
```

---

ICC profile tagging by value for TIFF image files is done by using the well-defined ICC private tag:

Tag number: 34675 (ICC assigned tag)  
Tag type: UNDEFINED

Format of the data is:

Embedded ICC profile.

Please refer to the Adobe TIFF documentation for further information on the Tagged Image File Format.

## Tagging for PDF files

ICC profile tagging by reference and value for PDF files is done by using a private dictionary. As soon as the ICC or Adobe define a common format for storing ICC profile information in PDF files, this format will be adapted by HELIOS.

The private dictionary 'HELInfo' is stored in the document's 'Info' dictionary:

<b>HELInfo</b>	Type:Dictionary	(Optional) All HELIOS private data is stored in the tree below this dictionary. Keys used in this dict:
<b>IccProfiles</b>	Type:Dictionary	(Optional) All ICC profile data is stored in this dictionary. Keys used in this dict:
<b>DeviceCMYK</b>	Type:String	A reference to a CMYK ICC profile. Format is the HELIOS ICC profile reference as described above.
	or: Stream	(Must be indirect) Embedded CMYK ICC profile.
<b>DeviceRGB</b>	Type:String	A reference to a RGB ICC profile. Format is the HELIOS ICC profile reference as described above.
	or: Stream	(Must be indirect) Embedded RGB ICC profile.

The following pages contain extracts from Adobe's and the ICC's original documentation about embedding profiles in EPSF and TIFF files:

**EPSF** (Adobe documentation)

Title: Embedding International Color Consortium Profiles in EPS

**TIFF** (ICC documentation)

Title: Embedding Device Profiles within Documents - TIFF



## Embedding International Color Consortium Profiles in EPS

There are two places within EPS files that embedding International Color Consortium (ICC) profiles are appropriate. 1) Associated with a screen preview. 2) Associated with the page description. Embedding ICC profiles within a screen preview is necessary so that applications using this screen preview to display a representation of the EPS page description can do so with accurate colors. Embedding ICC profiles within a page description is necessary so that sophisticated applications, such as OPI server software, can perform color conversions along with image replacement. For general information concerning PostScript's Document Structuring Conventions (DSC), the EPS file format, or specific PostScript operators, see the PostScript Language Reference Manual, second edition.

1) There are a variety of different methods of storing a screen preview within an EPS file depending on the intended environment. For cross platform applications with embedded ICC profiles, TIFF screen previews are recommended. The TIFF format has been extended to support the embedding of ICC profiles. ICC profiles can also be embedded in a platform specific manner. For example on the Macintosh, Apple has defined a method for embedding ICC profiles in PICT files.

Note that a given page description may use multiple distinct color spaces. In such cases, color conversions must be performed to a single color space to associate with the screen preview.

2) ICC profiles can also be embedded in the page description portion of an EPS file using the %%BeginICCPProfile / %%EndICCPProfile comments. This convention is defined as follows.

```
%%BeginICCPProfile: <profileid> <numberof> [<type> [<bytesorlines>]]
<profileid> ::= <text> (Profile ID)
<numberof> ::= <int> (Lines or physical bytes)
<type> ::= Hex | ASCII (Type of data)
<bytesorlines> ::= Bytes | Lines (Read in bytes or lines)
%%EndICCPProfile (no keywords)
```

These comments are designed to provide information about embedded ICC profiles. If the type argument is missing, ASCII data is assumed. ASCII refers to an ASCII base-85 representation of the data. If the bytesorlines argument is missing, <numberof> shall be considered to indicate bytes of data. If <numberof> = -1, the number of bytes of data are unknown. In this case, to skip over the profile one must read data until the encountering the %%EndICCPProfile comment.

<profileID> provides the profile's ID in order to synchronize it with PostScript's setcolorspace and findcolorrendering operators and associated operands (see below). Note that <numberof> indicates the bytes of physical data, which vary from the bytes of virtual data in some cases. With hex, each byte of virtual data is represented by two ASCII characters (two bytes of physical data). Although the PostScript interpreter ignores white space and percent signs in hex and ASCII data, these count toward the byte count.

Each line of profile data shall begin with a single percent sign followed by a space (% ). This makes the entire profile section a PostScript language comment so the file can be sent directly to a

printer without modification. The space avoids confusion with the open extension mechanism associated with DSC comments.

ICC profiles can be embedded within EPS files to allow sophisticated applications, such as OPI server software, to extract the profiles, and to perform color processing based on these profiles. In such situations it is desirable to locate the page description's color space and rendering intent, since this color space and rendering intent may need to be modified based on any color processing. The %%BeginSetColorSpace / %%EndSetColorSpace and %%BeginRenderingIntent / %%EndRenderingIntent comments are used to delimit the color space and rendering intent respectively.

```
%%BeginSetColorSpace <profileid>
  <profileid> ::= <text>          (ICC Profile ID)
%%EndSetColorSpace (no keywords)
```

<profileid> provides the ICC profile's ID corresponding to this color space. The ICC profile with this profile must have occurred in the PostScript job using the %%BeginICCProfile / %%EndICCProfile comment convention prior to this particular %%BeginSetColorSpace comment.

An example usage is shown here for CIE 1931 (XYZ)-space with D65 white point that refers to the ICC profile with <profileid> = XYZProfile.

```
%%BeginSetColorSpace XYZProfile
[/CIEBasedABC <<
  /WhitePoint [0.9505 1 1.0890]
  /RangeABC [0 0.9505 0 1 0 1.0890]
  /RangeLMN [0 0.9505 0 1 0 1.0890]
>>] setcolorspace
%%EndSetColorSpace
```

Note that the setcolorspace command is included within the comments. The PostScript enclosed in these comments shall not perform any other operations other than setting the color space and shall have no side effects.

```
%%BeginRenderingIntent <profileid>
  <profileid> ::= <text>          (ICC Profile ID)
%%EndRenderingIntent
```

<profileid> provides the ICC profile's ID corresponding to this rendering intent. The ICC profile with this profile must have occurred in the PostScript job using the %%BeginICCProfile / %%EndICCProfile comment convention prior to invocation of this particular %%BeginRenderingIntent comment.

An example usage is shown here for the the Perceptual rendering intent that refers to the ICC profile with <profileid> = RGBProfile.

```
%%BeginRenderingIntent RGBProfile
/Perceptual findcolorrendering pop
/ColorRendering findresource setcolorrendering
%%EndRenderingIntent
```

Note that the setcolorrendering command is included within the comments. The PostScript enclosed in these comments shall not perform any other

operations other than setting the rendering intent and shall have no side effects.

## 9 Embedding Device Profiles within Documents

This sections details the requirements and options for embedding device profiles within PICT, EPS and TIFF documents. All profiles except abstract profile can be embedded. The complete profile must be embedded with all tags intact and unchanged.

Embedding devicelink profiles renders the color data device dependent and significantly reduces portability. This may be useful in some situations, but may also cause problems with accurate color reproduction.

### 9.1 PICT

To be supplied by Apple.

### 9.2 EPS

To be supplied by Adobe.

### 9.3 TIFF

The discussion below assumes some familiarity with TIFF internal structure. It is beyond the scope of this document to detail the TIFF format, and readers are referred to the "TIFF(tm) Revision 6.0" specification, which is available from the Aldus Corporation.

The InterColor Consortium (IC) has been assigned a private TIFF tag for purposes of embedding IC device profiles within TIFF image files. This is not a required TIFF tag, and Baseline TIFF readers are not currently required to read it. It is, however, strongly recommended that this tag be honored.

A IC device profile is embedded, in its entirety, as a single TIFF field or Image File Directory (IFD) entry in the IFD containing the corresponding image data. An IFD should contain no more than one embedded profile. A TIFF file may contain more than one image, and so, more than one IFD. Each IFD may have its own embedded profile. Note, however, that Baseline TIFF readers are not required to read any IFDs beyond the first one.

The structure of the IC Profile IFD entry is as follows:

Bytes 0-1	The TIFF Tag that identifies the field = 34675(8773.H)
Bytes 2-3	The field Type = 7 = UNDEFINED (treated as 8-bit bytes).
Bytes 4-7 bytes.	The Count of values = the size of the embedded IC profile in bytes.
Bytes 8-11	The Value Offset = the file offset, in bytes, to the beginning of the IC profile.

Like all IFD entry values, the embedded profile must begin on a word boundary, so the Value Offset will always be an even number.

A TIFF reader should have no knowledge of the internal structure of an embedded

IC profile and should extract the profile intact.