

# Extract ICC Profile from Images

**Finnbarr P. Murphy**

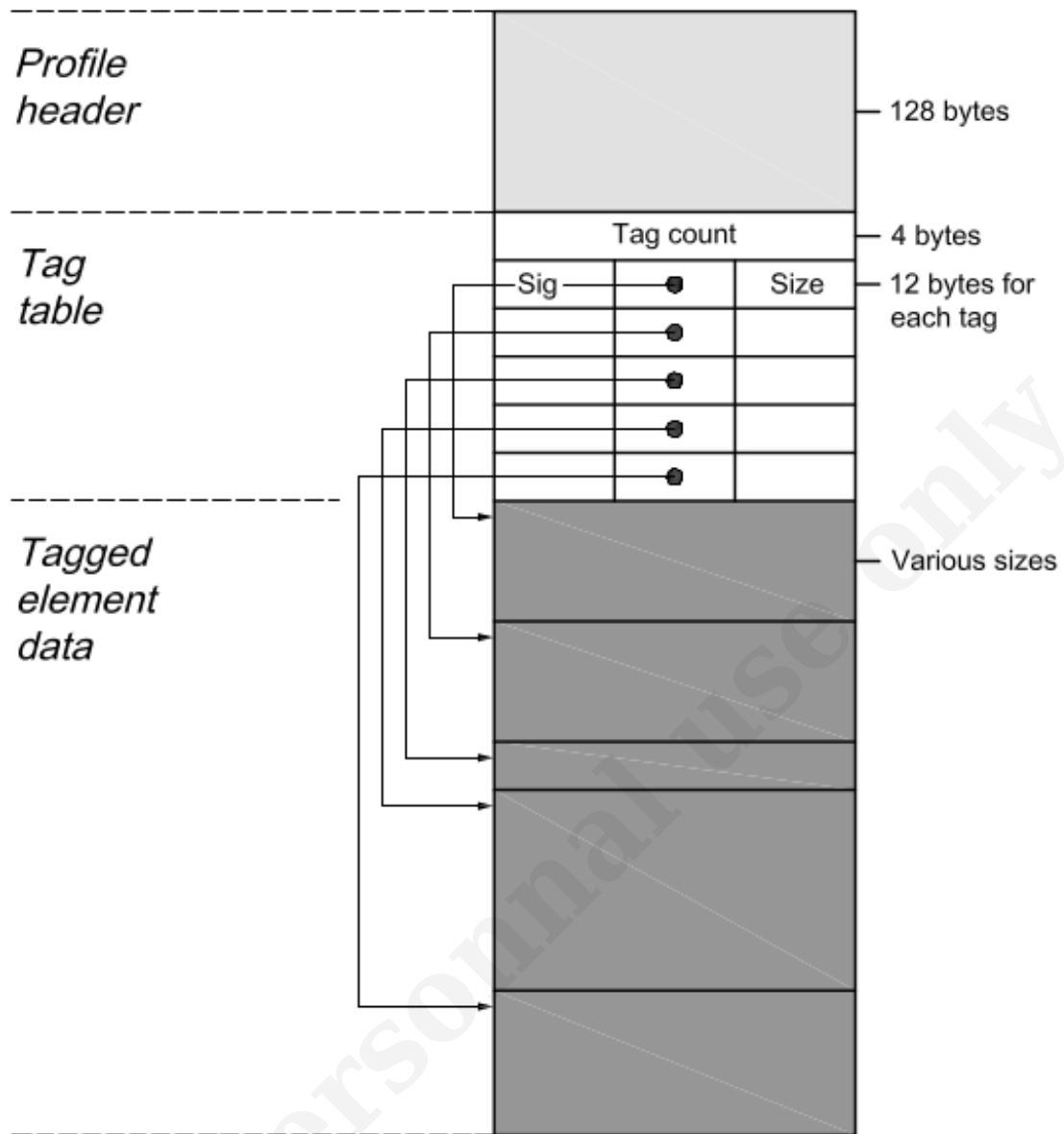
([fpm@fpmurphy.com](mailto:fpm@fpmurphy.com))

Color management is regarded as a black art by many application and system developers but it is becoming an increasingly important subject as color fidelity between devices is now a requirement for many applications.

What is color? Color (or *colour*) is the visual perceptual property corresponding in humans to the categories called red, green, blue, and others. Color derives from the spectrum of light (distribution of light power versus wavelength) interacting in the eye with the spectral sensitivities of the light receptors. Color categories and physical specifications of color are also associated with objects, materials, light sources, etc., based on their physical properties such as light absorption, reflection, or emission spectra. Color spaces, such as RGB, CYMK and [CIE 1931](#), represent colors as tuples of numbers. Other types of color spaces such as [RAL](#) are color matching schemes used by designers, interior decorators *et al.*

The [ICC \(International Color Consortium\) Profile Format](#) is an industry initiative to provide an interchange format to help solve the problems of specifying color, and in transferring color graphics from, and between systems and devices. The current version (ICC.1:2010 Profile Version 4.3.0.0, technically identical to ISO 15076-1:2010) of the specification enables color matching of images from the point of creation to the final output, whether display or print, in applications or within an operating system. Annex B, which is informative only, provides guidelines for embedding a profile in [EPS](#) (Encapsulated PostScript) or an image file. For example, in JPEG image files, the APP2 marker is used to introduce the ICC profile tag which, in turn, is identified by beginning the data with a special null terminated byte sequence, "ICC\_PROFILE".

As shown below, the ICC profile structure within an image is defined as a header followed by a tag table followed by a series of tagged elements (all defined in the specification) that can be accessed randomly and individually.



Within the profile structure:

- All profile data is encoded as big-endian.
- The first set of tagged element data immediately follows the tag table.
- Tagged element data is aligned on a 4-byte boundary using NULLs.

The profile header is intended to provide the necessary information to allow a receiving system to properly search and sort ICC profiles. It is 128 bytes in length and contains 18 fields. Here is a detailed breakdown:

Byte position	Field length bytes	Field contents
0 to 3	4	Profile size
4 to 7	4	Preferred CMM type
8 to 11	4	Profile version number
12 to 15	4	Profile/Device class
16 to 19	4	Colour space of data (possibly a derived space)
20 to 23	4	PCS
24 to 35	12	Date and time this profile was first created
36 to 39	4	'acsp' (61637370h) profile file signature
40 to 43	4	Primary platform signature
44 to 47	4	Profile flags to indicate various options for the CMM such as distributed processing and caching options
48 to 51	4	Device manufacturer of the device for which this profile is created
52 to 55	4	Device model of the device for which this profile is created
56 to 63	8	Device attributes unique to the particular device setup such as media type
64 to 67	4	Rendering Intent
68 to 79	12	The nCIEXYZ values of the illuminant of the PCS
80 to 83	4	Profile creator signature
84 to 99	16	Profile ID
100 to 127	28	Bytes reserved for future expansion and shall be set to zero (00h)

Note that the signature field always contains the value "acsp" (61637379h). I use that fact later to locate an ICC profile in an image.

The tag table provides a table of contents for the tagging information in each individual profile. The collection of tagged elements consist of three types: required data, optional data and private data.

Byte offset	Field length bytes	Content	Encoded as
0 to 3	4	Tag count ( $n$ )	
4 to 7	4	Tag Signature	
8 to 11	4	Offset to beginning of tag data element	uint32Number
12 to 15	4	Size of tag data element	uint32Number
16 to $(12n+3)$	$12(n-1)$	Signature, offset and size respectively of subsequent $n-1$ tags	
$n$ is the number of tags contained in the profile			

Each individual tag element includes a tag signature, the beginning address offset and size of the data. Signatures are defined as a 4-byte hexadecimal number as shown above. This design allows applications to read the element tag table and then load into memory only the information

necessary to their particular application. A detailed descriptions of the tags, along with their intent, are included in the ICC specification.

Here is a simple C utility (*iccdump*) which demonstrates one way of of extracting an ICC profile from an image, if one exists, and displaying it in a human-readable format.

```

/*
 * Copyright (c) 2012  Finnbarr P. Murphy.  All rights reserved
 *
 */
/*
 * Copyright 1997 - 2012 Graeme W. Gill
 *
 * This material is licensed with an "MIT" free use license:-
 */
#include <stdio.h>
#include <stdlib.h>
#include <stdarg.h>
#include <fcntl.h>
#include <string.h>
#include "icc.h"
#define MXTGNMS 30
void
error(char *fmt, ...)
{
    va_list args;
    fprintf(stderr,"ERROR: ");
    va_start(args, fmt);
    vfprintf(stderr, fmt, args);
    va_end(args);
    fprintf(stderr, "\n");
    exit(1);
}
void
usage(void) {
    fprintf(stderr,"usage: iccdump infile\n");
    exit(1);
}
int
main(int argc, char *argv[]) {
    int offset = 0;      /* Offset to read profile from */
    int found;
    icmFile *fp, *op;
    icc *icco;
    int rv = 0;
    if (argc < 2)
        usage();
    /* Open up the file for reading */
    if ((fp = new_icmFileStd_name(argv[1],"r")) == NULL)
        error("Cannot open file '%s'", argv[1]);
    if ((icco = new_icc()) == NULL)
        error("Creation of ICC object failed");
    /* open output stream */
    if ((op = new_icmFileStd_fp(stdout)) == NULL)
        error("Cannot open stdout stream");
    do {
        found = 0;
        /* Dumb search for magic number */
        int fc = 0;
        char c;
        if (fp->seek(fp, offset) != 0)
            break;
        while(found == 0) {
            if (fp->read(fp, &c, 1, 1) != 1)

```

```

        break;
    offset++;
    switch (fc) {
        case 0:
            if (c == 'a')
                fc++;
            else
                fc = 0;
            break;
        case 1:
            if (c == 'c')
                fc++;
            else
                fc = 0;
            break;
        case 2:
            if (c == 's')
                fc++;
            else
                fc = 0;
            break;
        case 3:
            if (c == 'p') {
                found = 1;
                offset -= 40;
            } else
                fc = 0;
            break;
    }
}
if (found) {
    printf("Embedded ICC profile found at file offset %d (0x%x)\n",offset,offset);
    if ((rv = icco->read(icco,fp,offset)) != 0)
        error("%d, %s", rv, icco->err);
    else
        icco->dump(icco, op, 3);
    offset += 128;
}
} while (found != 0);
icco->del(icco);
op->del(op);
fp->del(fp);
return 0;
}

```

This utility uses the excellent ICC profile parsing code developed by Graeme Gill as part of the [Argyll CMS](#) (Color Management System).

Here is sample output:

```

$ ./iccdump sample1.jpg
Embedded ICC profile found at file offset 6905 (0x1af9)
icc:
Header:
  size           = 3144 bytes
  CMM            = 'Lino'
  Version        = 2.1.0
  Device Class   = Display
  Color Space    = RGB
  Conn. Space    = XYZ
  Date, Time     = 9 Feb 1998, 6:49:00
  Platform       = Microsoft
  Flags          = Not Embedded Profile, Use anywhere
  Dev. Mnfctr.   = 'IEC '

```

## Extract ICC Profile from Images

```
Dev. Model = 'sRGB'  
Dev. Attribts = Reflective, Glossy, Positive, Color  
Rndrng Intnt = Perceptual  
Illuminant = 0.964203, 1.000000, 0.824905 [Lab 100.000000, 0.000498, -0.000436]  
Creator = 'HP '
```

```
tag 0:  
sig 'cprt'  
type 'text'  
offset 336  
size 51
```

```
Text:  
No. chars = 43  
0x0000: Copyright (c) 1998 Hewlett-Packard Company
```

```
tag 1:  
sig 'desc'  
type 'desc'  
offset 388  
size 108
```

```
TextDescription:  
ASCII data, length 18 chars:  
0x0000: sRGB IEC61966-2.1  
No Unicode data  
ScriptCode Data, Code 0x0, length 18 chars  
0x0000: 73 52 47 42 20 49 45 43 36 31 39 36 36 2d 32 2e 31 00
```

```
tag 2:  
sig 'wtpt'  
type 'XYZ '  
offset 496  
size 20
```

```
XYZArray:  
No. elements = 1  
0: 0.950455, 1.000000, 1.089050 [Lab 100.000000, -2.387320, -19.404505]
```

```
tag 3:  
sig 'bkpt'  
type 'XYZ '  
offset 516  
size 20
```

```
XYZArray:  
No. elements = 1  
0: 0.000000, 0.000000, 0.000000 [Lab 0.000000, 0.000000, 0.000000]
```

```
tag 4:  
sig 'rXYZ'  
type 'XYZ '  
offset 536  
size 20
```

```
XYZArray:  
No. elements = 1  
0: 0.436066, 0.222488, 0.013916 [Lab 54.290039, 80.819767, 69.895569]
```

```
tag 5:  
sig 'gXYZ'  
type 'XYZ '  
offset 556  
size 20
```

```
XYZArray:  
No. elements = 1  
0: 0.385147, 0.716873, 0.097076 [Lab 87.817866, -79.257408, 80.986979]
```

```
tag 6:  
sig 'bXYZ'  
type 'XYZ '  
offset 576  
size 20
```

```
XYZArray:
  No. elements = 1
    0: 0.143066, 0.060608, 0.714096 [Lab 29.565320, 68.301563, -112.050315]

tag 7:
  sig 'dmnd'
  type 'desc'
  offset 596
  size 112
TextDescription:
  ASCII data, length 22 chars:
    0x0000: IEC http://www.iec.ch
  No Unicode data
  ScriptCode Data, Code 0x0, length 22 chars
    0x0000: 49 45 43 20 68 74 74 70 3a 2f 2f 77 77 77 2e 69 65 63 2e 63 68 00

tag 8:
  sig 'dmdd'
  type 'desc'
  offset 708
  size 136
TextDescription:
  ASCII data, length 46 chars:
    0x0000: IEC 61966-2.1 Default RGB colour space - sRGB
  No Unicode data
  ScriptCode Data, Code 0x0, length 46 chars
    0x0000: 49 45 43 20 36 31 39 36 36 2d 32 2e 31 20 44 65 66 61 75 6c 74 20
    0x0016: 52 47 42 20 63 6f 6c 6f 75 72 20 73 70 61 63 65 20 2d 20 73 52 47
    0x002c: 42 00

tag 9:
  sig 'vued'
  type 'desc'
  offset 844
  size 134
TextDescription:
  ASCII data, length 44 chars:
    0x0000: Reference Viewing Condition in IEC61966-2.1
  No Unicode data
  ScriptCode Data, Code 0x0, length 44 chars
    0x0000: 52 65 66 65 72 65 6e 63 65 20 56 69 65 77 69 6e 67 20 43 6f 6e 64
    0x0016: 69 74 69 6f 6e 20 69 6e 20 49 45 43 36 31 39 36 36 2d 32 2e 31 00

tag 10:
  sig 'view'
  type 'view'
  offset 980
  size 36
Viewing Conditions:
  XYZ value of illuminant in cd/m^2 = 19.644501, 20.371796, 16.808899
  XYZ value of surround in cd/m^2 = 3.928894, 4.074387, 3.361786
  Illuminant type = D50

tag 11:
  sig 'lumi'
  type 'XYZ '
  offset 1016
  size 20
XYZArray:
  No. elements = 1
    0: 76.036469, 80.000000, 87.124619 [Lab 483.828848, -10.285791, -83.613628]

tag 12:
  sig 'meas'
  type 'meas'
  offset 1036
  size 36
Measurement:
```

```
Standard Observer = 1931 Two Degrees
XYZ for Measurement Backing = 0.000000, 0.000000, 0.000000 [Lab 0.000000, 0.000000, 0.000000]
Measurement Geometry = Unknown
Measurement Flare = 1.0%
Standard Illuminant = D65

tag 13:
sig      'tech'
type     'sig '
offset   1072
size     12
Signature
Technology = Cathode Ray Tube Display

tag 14:
sig      'rTRC'
type     'curv'
offset   1084
size     2060
Curve:
No. elements = 1024
0: 0.000000
1: 0.000076
2: 0.000153
3: 0.000229
4: 0.000305
5: 0.000381
6: 0.000458
.....
1020: 0.993347
1021: 0.995560
1022: 0.997772
1023: 1.000000

tag 15:
sig      'gTRC'
type     'curv'
offset   1084
size     2060
Curve:
No. elements = 1024
0: 0.000000
1: 0.000076
2: 0.000153
3: 0.000229
4: 0.000305
5: 0.000381
6: 0.000458
.....
1020: 0.993347
1021: 0.995560
1022: 0.997772
1023: 1.000000

tag 16:
sig      'bTRC'
type     'curv'
offset   1084
size     2060
Curve:
No. elements = 1024
0: 0.000000
1: 0.000076
```



```

2: 0.000153
3: 0.000229
4: 0.000305
5: 0.000381
6: 0.000458

.....

1020: 0.993347
1021: 0.995560
1022: 0.997772
1023: 1.000000
    
```

And here is an example of a simpler profile:

```

$ ./iccdump sample2.jpg

Embedded ICC profile found at file offset 9279 (0x243f)
icc:
Header:
  size           = 560 bytes
  CMM             = 'ADBE'
  Version        = 2.1.0
  Device Class   = Display
  Color Space    = RGB
  Conn. Space    = XYZ
  Date, Time     = 3 Jun 1999, 0:00:00
  Platform       = Macintosh
  Flags          = Not Embedded Profile, Use anywhere
  Dev. Mnfrctr. = 'none'
  Dev. Model     = 0x0
  Dev. Attribts = Reflective, Glossy, Positive, Color
  Rndrng Intnt  = Perceptual
  Illuminant     = 0.964203, 1.000000, 0.824905 [Lab 100.000000, 0.000498, -0.000436]
  Creator       = 'ADBE'

tag 0:
  sig    'cprt'
  type   'text'
  offset 252
  size   50
Text:
  No. chars = 42
  0x0000: Copyright 1999 Adobe Systems Incorporated

tag 1:
  sig    'desc'
  type   'desc'
  offset 304
  size   107
TextDescription:
  ASCII data, length 17 chars:
  0x0000: Adobe RGB (1998)
  No Unicode data
  No ScriptCode data

tag 2:
  sig    'wtpt'
  type   'XYZ '
  offset 412
  size   20
XYZArray:
  No. elements = 1
  0: 0.950455, 1.000000, 1.089050 [Lab 100.000000, -2.387320, -19.404505]
    
```

```
tag 3:
  sig      'bkpt'
  type     'XYZ '
  offset   432
  size     20
XYZArray:
  No. elements = 1
    0: 0.000000, 0.000000, 0.000000 [Lab 0.000000, 0.000000, 0.000000]

tag 4:
  sig      'rTRC'
  type     'curv'
  offset   452
  size     14
Curve:
  Curve is gamma of 2.199219

tag 5:
  sig      'gTRC'
  type     'curv'
  offset   468
  size     14
Curve:
  Curve is gamma of 2.199219

tag 6:
  sig      'bTRC'
  type     'curv'
  offset   484
  size     14
Curve:
  Curve is gamma of 2.199219

tag 7:
  sig      'rXYZ'
  type     'XYZ '
  offset   500
  size     20
XYZArray:
  No. elements = 1
    0: 0.609741, 0.311111, 0.019470 [Lab 62.601347, 90.371212, 78.149349]

tag 8:
  sig      'gXYZ'
  type     'XYZ '
  offset   520
  size     20
XYZArray:
  No. elements = 1
    0: 0.205276, 0.625671, 0.060867 [Lab 83.214105, -129.089932, 87.172524]

tag 9:
  sig      'bXYZ'
  type     'XYZ '
  offset   540
  size     20
XYZArray:
  No. elements = 1
    0: 0.149185, 0.063217, 0.744568 [Lab 30.210038, 69.243738, -113.612302]
```

The entire source code and a Makefile is available under the *iccdump* directory the *Various* repository on [GitHub](#).

Enjoy!

For personal use only