Getting Metadata from Digital Photographs using C#

John F. Gnazzo

Introduction

Contemporary digital cameras imprint metadata, or information about the photo, within the photo itself. This article will discuss what information is commonly available, specific data extracted from a specific JPEG image, and how to extract it using the C# programming language.

Digital Photograph Metadata

Several types of data can be extracted from the image such as property information and compositional elements.

The following is a standard JPEG formatted image taken from a Nikon 3100 Digital Camera.

The following table shows what image property information is commonly available, and present in the above image.

The table includes the following information:

- Item ID in decimal format
- Item ID in hexagonal format
- Type ID

The complete list of Type IDs are included in the following link:

https://msdn.microsoft.com/en-us/library/system.drawing.imaging.propertyitem.type(v=vs.110).aspx

- Data type

  The following is a list of data types that can be returned:

  - Byte Array
  - String
  - Unsigned Int32 Array
  - Unsigned Int64 Array
  - Signed Int32 Array
  - Signed Int64 Array
  - Rational (Array of Pairs of Unsigned Int64)
  - SRational (Array of Pairs of Signed Int64)
- Property

  The complete list of properties that can be in an image are included in the following links.

  https://msdn.microsoft.com/de-de/library/ms534416.aspx

  - http://exif-utils.googlecode.com/svn/trunk/ExifUtils/ExifUtils/Exif/ExifTag.cs
- Value

Item ID Decimal	Item ID Hex	Type ID	Data Type	Property	Value
271	0x010f	2	ASCII	EquipMake	NIKON CORPORATION
272	0x0110	2	ASCII	EquipModel	NIKON D3100
274	0x0112	3	Short (16 bit int)	Orientation	1
282	0x011a	5	Rational	XResolution	300,1
283	0x011b	5	Rational	YResolution	300,1
296	0x0128	3	Short (16 bit int)	ResolutionUnit	2
305	0x0131	2	ASCII	SoftwareUsed	Ver.1.01
306	0x0132	2	ASCII	DateTime	2014:04:19 12:33:40
531	0x0213	3	Short (16 bit int)	YCbCrPositioning	2
33434	0x829a	5	Rational	ExifExposureTime	1,012,500
33437	0x829d	5	Rational	ExifFNumber	56,10
34850	0x8822	3	Short (16 bit int)	ExifExposureProg	0
34855	0x8827	3	Short (16 bit int)	ExifISOSpeed	400
36864	0x9000	7	Undefined	ExifVer	
36867	0x9003	2	ASCII	ExifDTOrig	2014:04:19 12:33:40
36868	0x9004	2	ASCII	ExifDTDigitized	2014:04:19 12:33:40
37121	0x9101	7	Undefined	ExifCompConfig	

37122	0x9102	5	Rational	ExifCompBPP	2,1
37380	0x9204	10	SRational	ExifExposureBias	0,6
37381	0x9205	5	Rational	ExifMaxAperture	47,10
37383	0x9207	3	Short (16 bit int)	ExifMeteringMode	5
37384	0x9208	3	Short (16 bit int)	ExifLightSource	0
37385	0x9209	3	Short (16 bit int)	ExifFlash	16
37386	0x920a	5	Rational	ExifFocalLength	1850,10
37500	0x927c	7	Undefined	ExifMakerNote	
37510	0x9286	7	Undefined	ExifUserComment	
37520	0x9290	2	ASCII	ExifDTSubsec	80
37521	0x9291	2	ASCII	ExifDTOrigSS	80
37522	0x9292	2	ASCII	ExifDTDigSS	80
40960	0xa000	7	Undefined	ExifFPXVer	
40961	0xa001	3	Short (16 bit int)	ExifColorSpace	1
40962	0xa002	3	Short (16 bit int)	ExifPixXDim	4608
40963	0xa003	3	Short (16 bit int)	ExifPixYDim	3072
20545	0x5041	2	ASCII	InteroperabilityIndex	R98
20546	0x5042	7	Undefined	Unknown	
41495	0xa217	3	Short (16 bit int)	ExifSensingMethod	2
41728	0xa300	7	Undefined	ExifFileSource	
41729	0xa301	7	Undefined	ExifSceneType	
41730	0xa302	7	Undefined	ExifCfaPattern	
41985	0xa401	3	Short (16 bit int)	CustomRendered	0
41986	0xa402	3	Short (16 bit int)	ExposureMode	0
41987	0xa403	3	Short (16 bit int)	WhiteBalance	0
41988	0xa404	5	Rational	DigitalZoomRatio	1,1
41989	0xa405	3	Short (16 bit int)	FocalLengthIn35mmFilm	277
41990	0xa406	3	Short (16 bit int)	SceneCaptureType	0
41991	0xa407	3	Short (16 bit int)	GainControl	1
41992	0xa408	3	Short (16 bit int)	Contrast	0
41993	0xa409	3	Short (16 bit int)	Saturation	0
41994	0xa40a	3	Short (16 bit int)	Sharpness	0
41996	0xa40c	3	Short (16 bit int)	SubjectDistanceRange	0
0	0x0	1	Array of bytes	GpsVer	2,2,0,0
20507	0x501b	1	Array of bytes	ThumbnailData	values
20515	0x5023	3	Short (16 bit int)	ThumbnailCompression	6
20525	0x502d	5	Rational	ThumbnailResolutionX	300,1
20526	0x502e	5	Rational	ThumbnailResolutionY	300,1
20528	0x5030	3	Short (16 bit int)	ThumbnailResolutionUnit	2
513	0x201	4	Long (32 bit int)	JPEGInterFormat	36624
514	0x202	4	Long (32 bit int)	JPEGInterLength	9041
20537	0x5039	3	Short (16 bit int)	ThumbnailYCbCrPositioning	2

20625	0x5091	3	Short (16 bit int)	ChrominanceTable	values
20624	0x5090	3	Short (16 bit int)	LuminanceTable	values

The following table shows what compositional elements are available for an image, and their values for the image above.

Element	Value	Description
Flags	77840	A bitwise combination of flags that denote such items as the images color space, and pixel data
FrameDimensionList	Guid[]	1 or more frames in the image
Height	3072	Height of the image in pixels
Width	4608	Width of the image in pixels
Horizontal Resolution	300	Pixels per inch in the horizontal dimension
Vertical Resolution	300	Pixels per inch in the vertical dimension
Palette	ColorPalette,	Color pallet used and if the image uses Alpha, Greyscale, and/or halftone
	6488174	
PixelFormat	Format24bppRgb	Format of the pixel
RawFormat	jpg	File Format of the image

Extracting the Metadata using C#

Extracting photo meta data is easy using C#. The following function is an example of how to extract meta data from a photograph.

This method extracts the **Picture Taken Date** from the photograph passed to the method as a Bitmap object.

From the above table the id 36867 is the item id for the Date Time of the photo.

```
/// <summary>
/// Gets the picture taken date from bitmap.
/// </summary>
/// <param name="bitmap">the phto</param>
/// <returns></returns>
private static DateTime? GetPictureTakenDateFromBitmap(Bitmap bitmap)
{
    var r = new Regex(":");
    try
    {
        // 36867 is the item number of the picture taken date.
        var MyItem = bitmap.GetPropertyItem(36867);

        string date = r.Replace(Encoding.UTF8.GetString(MyItem.Value), "-", 2);
```

```
            return Convert.ToDateTime(date);
        }
        catch (Exception)
        {
            return null;
        }
    }
```

This method could easily be abstracted to return any property as a string using the following method signature:

```
private static string GetDataFromBitmap(Bitmap bitmap, int typeId)
```

Conclusion

This article discussed the information that is commonly available in a digital photograph, specific data extracted from a specific JPEG image, and how to extract the data it using the C# programming language.