

Registry support for multimedia and metadata in EMu 3.2.03.

- [Overview](#)
- [ImageMagick](#)
- [Multimedia](#)
 - [Audio](#)
 - [Video](#)
 - [Images](#)
- [Metadata](#)
 - [EXIF](#)
 - [IPTC](#)
 - [XMP](#)
 - [Embed in derivatives](#)
 - [Extract into Multimedia module](#)
 - [Limiting Colours computation](#)

Overview

The image, audio and video libraries used to support multimedia have been replaced in KE EMu 3.2.03. The previous libraries were becoming dated and lacked support for newer file formats, in particular 16 bit graphics and CMYK colour spaces, as well as JPEG 2000. The previous libraries also used a simple algorithm for resizing images, which led to loss of clarity and colour.

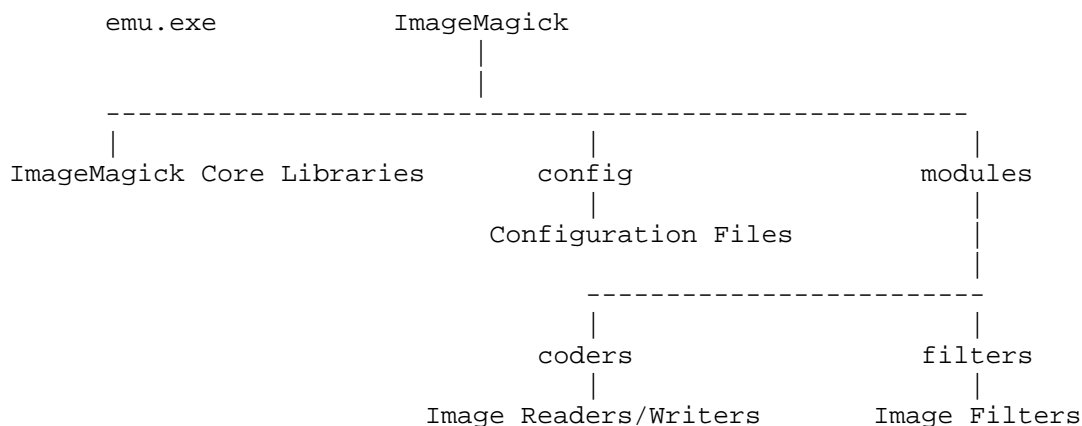
Rather than tie EMu image development to a third party vendor an open source solution was adopted as this provides development for new image formats and metadata standards as they emerge. It was decided that [ImageMagick](#) offered the functionality to expand the current image support in EMu. Unfortunately ImageMagick does not provide support for audio or video formats, so it was decided to build this functionality into EMu rather than use third party libraries.

Access to metadata stored in image files is made available through ImageMagick. In particular, it has limited support for EXIF, IPTC and XMP metadata profiles. EMu now uses this support to extract metadata from master images and to embed metadata into derived images.

This document describes how the new multimedia and metadata features can be configured using the EMu Registry.

ImageMagick

The ImageMagick libraries distributed with EMu are stored under the same directory as the program executable. If a network installation is performed, the libraries reside on a server machine in a directory accessible to all client machines. In the case of a standalone installation, they are installed on the local machine. The libraries do not require any registration with Windows (that is, they are not COM objects). The directory structure is:



where the ImageMagick Core Libraries provide the core image functionality (e.g. resizing code, etc.) and Configuration Files contains language specific text and other configuration files. These files do not need to be modified. Image Reader/Writers is a collection of files used to read and write specific image formats. For example, IM_MOD_RL_jpeg.dll is used to read and write JPEG files. Image Filters are used to provide statistical information about images (not used by EMu).

EMu installs the 16 bit ImageMagick libraries. These libraries are capable of storing 16 bits of colour information per colour channel: i.e. for RGB images 48 bits per pixel and for CMYK 64 bits per pixel. Most images produced until recently only store 8 bits of colour information per colour channel. Newer digital devices are now capable of producing 16 bit images that give a greater colour range and so result in “truer” colour pictures. It is to cater for these images that EMu distributes the 16 bit libraries. Note that 8 bit images can be read and manipulated by the 16 bit libraries without any loss of definition.

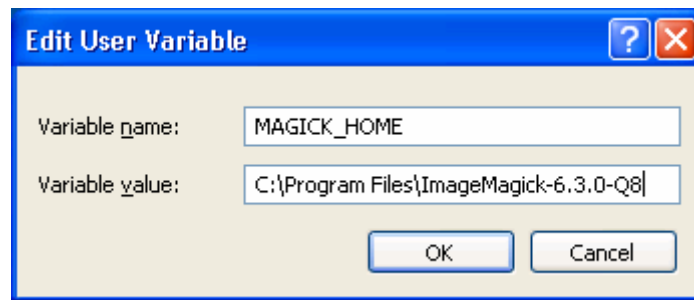
There is however an overhead associated with providing support for 16 bit images. When ImageMagick reads an image it decodes the file format (via a *coder*) and stores it in an uncompressed pixel format in memory. Each pixel uses 64 bits or 8 bytes **regardless** of the number of bits per pixel in the image file. If you read an image which is 1000 pixels by 1000 pixels, the memory required to hold the image is 8 Mb (1000 * 1000 * 8 bytes). Most computers these days can spare 8 Mb of memory to read an image. If however you are reading a 5000 pixel by 5000 pixel image, 200 Mb is required (5000 * 5000 * 8 bytes) even if it is a monochrome image! I imagine that unless you have at least 1 Gb of memory installed, your computer will slow down considerably (technically it will start swapping) when dealing with such images. So if you plan to manipulate large images, you may want to determine whether you have sufficient memory in your machine to manipulate the images in memory.

When the ImageMagick libraries are required for the first time, EMu loads them dynamically. This means that EMu determines where the libraries are installed (in the same directory as the EMu executable by default) and having done so “loads” them into memory. Once the libraries are loaded EMu can make use of them. In order to locate the libraries EMu first looks up the **MAGICK_HOME** environment variable. If it is set, EMu will attempt to load the ImageMagick libraries from the directory specified by the environment variable. If the environment variable is not set, EMu will look under the executable directory.

It is possible to take advantage of the MAGICK_HOME environment variable to force EMu to load a different version of the ImageMagick libraries than the ones distributed with EMu. If you want to halve your memory usage and also load images 33% faster, you can use the 8 bit ImageMagick libraries. By doing so however, you will not be able to view or read 16 bit images. To bypass the libraries distributed with EMu you need to:

1. Install the required version of ImageMagick on our local machine or on a network server. You can download the releases from <http://www.imagemagick.org/script/download.php>. Be sure to download the DLL version, not the static version.
2. Run *Start->Control Panel->System* and select the *Advanced* tab. Click the *Environment Variables* button.
3. Under *User variables for ...* click the *New* button.

4. Enter **MAGICK_HOME** for the *Variable name* and set the *Variable value* to the directory into which ImageMagick was installed:

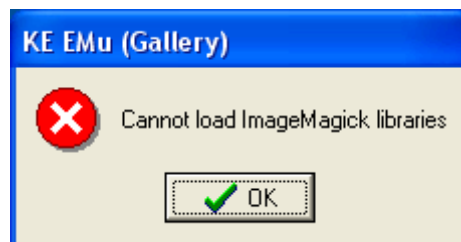


5. Click the *OK* button to save the setting.

Now, when EMu loads the ImageMagick libraries it will load those under the directory specified above. You can use the above technique to install later releases of the libraries that may support more image formats (although the libraries distributed with EMu will be updated on a regular basis).

If your institution does not want to use the 16 bit libraries and you have a network installation, you can copy the contents of the 8 bit image directory over the ImageMagick directory under the EMu executable directory. Using this technique you do not need to set the **MAGICK_HOME** environment variable. Note however that the next upgrade you perform will replace your libraries with the 16 bit versions, so you will need to perform the copy again.

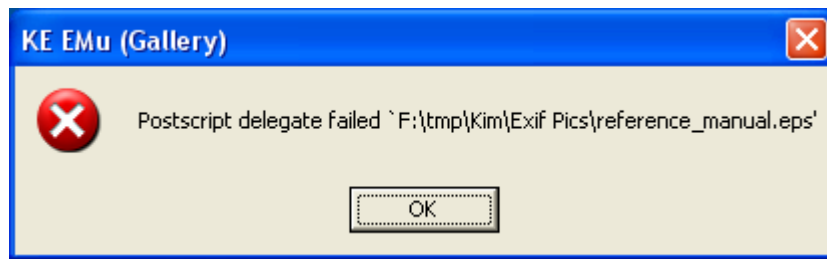
If EMu cannot load the ImageMagick libraries, the following error dialogue is displayed:



If you have set the **MAGICK_HOME** environment variable, this error indicates that one of the following conditions is not true:

- The environment variable value must contain the path to an existing directory.
- As a minimum the user must have the ability to list files in the specified directory.
- A valid set of ImageMagick libraries, including a **config** and **modules** directory and a series of DLL files must exist in the specified directory.
- All DLLs in the specified directory and **modules/coders** must have *Read* and *Execute* permission for the current user.
- As a minimum all directories below the specified directory (namely **config**, **modules**, **modules/coders** and **modules/filters**) must have *List Files* permission.
- As a minimum all configuration files in **config** must have *Read* permission.

ImageMagick uses *delegates* to provide support for certain image formats. A *delegate* is a piece of software that converts the image format into a format that ImageMagick can handle. EMu bundles two delegates with the ImageMagick libraries: **dcraw.exe** (handles digital camera RAW format) and **hp2xx.exe** (handles HPGL files). If you try to import or generate an image format that requires a delegate and the delegate is not installed, an error dialogue similar to this one is displayed:



In order to determine the delegate required and to download the necessary software, visit the [ImageMagick Formats](#) page. Once the delegate software is installed the image format will be useable.

Multimedia

EMu 3.2.03 sees the complete replacement of the image, audio and video sub-systems. The new sub-systems are designed to be self configuring and extensible. In order to determine what media formats are supported by a given machine, EMu determines what audio and video drivers are installed and queries ImageMagick to see what image formats are available. This self-configuration means that if you install software that supports a given media format, EMu can take advantage of this and where possible provide internal handling for the format.

The sections below provide details on how the audio, video and image sub-systems can be tailored to suit the requirements of a given institution.

Audio

The new audio sub-system uses the Windows Media Control Interface (MCI) to locate audio codec drivers (codec = coder/decoder) used to play audio files. To find the list of drivers installed on your computer:

1. Select *Start->Control Panel->Sounds and Audio Device*.
2. Click the *Hardware* tab.
3. Highlight the *Audio Codecs* device and click *Properties*.
4. Click the *Properties* tab. The tab contains the list of audio codecs installed.

When EMu first requires a list of audio files supported it fetches the list of installed drivers and makes each driver's file suffix available. A standard Windows XP machine will support the following file extensions:

- AIF - Audio Interchange File Format
- AIFC - Compressed Audio Interchange File
- AIFF - Audio Interchange File Format
- AU - Audio File Forma
- M3U - MP3 Playlist
- MID - Musical Instrument Digital Interface
- MP3 - MPEG Audio Stream, Layer III
- RMI - MIDI in RIFF File Format
- SND - Sound File Format
- WAV - Waveform Audio
- WAX - Windows Media Audio Redirect
- WMA - Windows Media Audio File

When a user wants to add a media file into the Multimedia Repository the dialogue box used to select the required media file will include all of the above file extensions under the *Audio Files* **File of type:** drop list.

Some institutions may want to restrict the list of file extensions available under the *Audio Files* list to formats approved for use. A Registry entry has been added that allows system administrators to limit the list of allowed audio extensions. The format of the entry is:

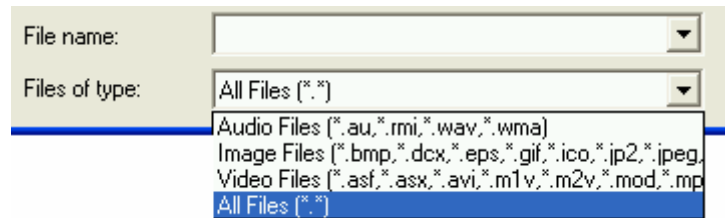
```
User | user | Setting | Multimedia | Audio | Exclude | format; ...
Group | group | Setting | Multimedia | Audio | Exclude | format; ...
Group | Default | Setting | Multimedia | Audio | Exclude | format; ...
System | Setting | Multimedia | Audio | Exclude | format; ...
```

where *format* is one of the file extensions listed above. The entry allows you to define formats that should not appear in the *Audio Files* list. Note that dropping an extension from the list does not stop users from selecting and importing audio files for that extension, rather it informs EMu that the file is not to be treated as an audio file.

As an example, let's say your institution only wants AU, WAV, WMA and RMI sound files to be used. The following entry could be used to remove the unwanted formats:

```
System|Setting|Multimedia|Audio|Exclude|aif;aifc;aiff;m3u;mid;mp3;snd;wax
```

which would result in the following *Audio Files* list:



For selected users you can provide a different list of audio extensions by using the User or Group versions of the Registry entry. If users in group **Admin** can also add MP3 and SND files, the following additional entry is required:

```
Group|Admin|Setting|Multimedia|Audio|Exclude|aif;aifc;aiff;m3u;mid;wax
```

Video

The new video sub-system also uses the Windows Media Control Interface (MCI) to locate video codec drivers used to play video files. The list of drivers installed on your computer can be found in the same way as for audio files except that the *Video Codecs* device is used. As with audio drivers, EMu fetches the list of installed video drivers and makes all file extensions available. A standard Windows XP machine will support the following file extensions:

- ASF - Advanced Streaming Format
- ASX - ASF Redirector File
- AVI - Audio Video Interleave File
- M1V - MPEG-1 Video File
- M2V - MPEG-2 Video File
- MOD - JVC Everio GZ-MG20U Digital Video File
- MP2 - MPEG Layer II Compressed Audio File
- MP2V - MPEG Audio Stream, Layer II
- MPA - MPEG Audio Stream, Layer I, II or III
- MPE - MPEG Movie Clip
- MPEG - MPEG 1 System Stream
- MPG - MPEG 1 System Stream
- MPv2 - MPEG Audio Stream, Layer II
- WM - Windows Media A/V File
- WMV - Windows Media File
- WMX - Windows Media Player A/V Shortcut
- WVX - Windows Media Redirector

The following Registry entries can be used to limit the *Video Files* list displayed when media is being added:

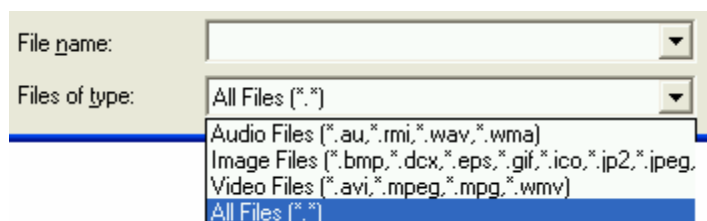
```
User|user|Setting|Multimedia|Video|Exclude|format;...
Group|group|Setting|Multimedia|Video|Exclude|format;...
Group|Default|Setting|Multimedia|Video|Exclude|format;...
System|Setting|Multimedia|Video|Exclude|format;...
```

where *format* is one of the extensions listed above. Again, if an extension is excluded from the *Video Files* list, it does not stop users from adding the file type. If a file type is excluded, EMu will treat it as a normal media file (in the same way that it treats Word and other document based file formats).

As an example, let's say your institution only wants MPG, MPEG, AVI and WMV video files to be used. The following entry could be used:

```
System | Setting | Multimedia | Video | Exclude | asf ; asx ; m1v ; m2v ; mod ; mp2 ; mp2v ; mpa ; mpe ; mpv2 ; wm ; wmx ; wvx
```

which would result in the following *Video Files* list:



Images

ImageMagick provides support for a large number of image formats. The actual list of file types supported depends on the version of ImageMagick installed. When EMu needs to display a list of supported image formats (either for adding an image or generating a derivative) it asks ImageMagick for a list of file formats that have coders. EMu then consults the Windows registry to determine whether each entry in the list has a MIME type of **image**, discarding entries that do not. Thus if you have an application installed that handles a given image format, EMu will add it to the *Image Files* list. Finally, the image format must have an encoder (used to write the format) and a decoder (used to read the format).

The list of image formats supported is:

File Extension	Coder	Read/Write/Multi-page	Format
A	RAW	rw+	Raw alpha samples
AVS	AVS	rw+	AVS X image
B	RAW	rw+	Raw blue samples
BIE	JBIG	rw-	Joint Bi-level Image experts Group interchange format (1.5)
BMP	BMP	rw-	Microsoft Windows bitmap image
C	RAW	rw+	Raw cyan samples
CIN	CIN	rw+	Cineon Image File
CMYK	CMYK	rw+	Raw cyan, magenta, yellow, and black samples
CMYKA	CMYK	rw+	Raw cyan, magenta, yellow, black, and opacity samples
CUR	CUR	rw-	Microsoft icon
CUT	CUT	r--	DR Halo
DCM	DCM	r--	Digital Imaging and Communications in Medicine image
DCX	PCX	rw+	ZSoft IBM PC multi-page Paintbrush
DNG	DNG	r--	Digital Negative
EMF	WMF	r--	Windows WIN32 API rendered Enhanced Meta File

File Extension	Coder	Read/ Write/ Multi-page	Format
EPDF	PDF	rw-	Encapsulated Portable Document Format
EPI	PS	rw-	Encapsulated PostScript Interchange format
EPS	PS	rw-	Encapsulated PostScript
EPSF	PS	rw-	Encapsulated PostScript
EPSI	PS	rw-	Encapsulated PostScript Interchange format
EPT	EPT	rw-	Encapsulated PostScript with TIFF preview
EPT2	EPT	rw-	Encapsulated PostScript Level II with TIFF preview
EPT3	EPT	rw+	Encapsulated PostScript Level III with TIFF preview
FAX	FAX	rw+	Group 3 FAX
FITS	FITS	rw-	Flexible Image Transport System
FTS	FTS	rw-	Flexible Image Transport System
G	RAW	rw+	Raw green samples
G3	FAX	rw-	Group 3 FAX
GIF	GIF	rw+	CompuServe graphics interchange format
GIF87	GIF	rw-	CompuServe graphics interchange format (version 87a)
GRAY	GRAY	rw+	Raw gray samples
ICB	TGA	rw+	Truevision Targa image
ICO	ICON	rw-	Microsoft icon
ICON	ICON	rw-	Microsoft icon
JBG	JBIG	rw+	Joint Bi-level Image experts Group interchange format (1.5)
JBIG	JBIG	rw+	Joint Bi-level Image experts Group interchange format (1.5)
JNG	PNG	rw-	JPEG Network Graphics
JP2	JP2	rw-	JPEG-2000 File Format Syntax
JPC	JPC	rw-	JPEG-2000 Code Stream Syntax
JPEG	JPEG	rw-	Joint Photographic Experts Group JFIF format (62)
JPG	JPEG	rw-	Joint Photographic Experts Group JFIF format
JPX	JPX	rw-	JPEG-2000 File Format Syntax
K	RAW	rw+	Raw black samples
M	RAW	rw+	Raw magenta samples
MAP	MAP	rw-	Colormap intensities and indices
MAT	MAT	rw+	MATLAB image format
MIFF	MIFF	rw+	Magick Image File Format
MNG	PNG	rw+	Multiple-image Network Graphics (libpng 1.2.8)
MONO	MONO	rw-	Bi-level bitmap in least-significant-byte first order
MPC	MPC	rw+	Magick Persistent Cache image format
MSL	MSL	rw+	Magick Scripting Language
MTV	MTV	rw+	MTV Raytracing image format

File Extension	Coder	Read/ Write/ Multi-page	Format
MVG	MVG	rw-	Magick Vector Graphics
O	RAW	rw+	Raw opacity samples
OTB	OTB	rw-	On-the-air bitmap
PAL	UYVY	rw-	16bit/pixel interleaved YUV
PALM	PALM	rw+	Palm pixmap
PAM	PNM	rw+	Common 2-dimensional bitmap format
PBM	PNM	rw+	Portable bitmap format (black and white)
PCD	PCD	rw-	Photo CD
PCDS	PCD	rw-	Photo CD
PCL	PCL	rw-	Printer Control Language
PCT	PICT	rw-	Apple Macintosh QuickDraw/PICT
PCX	PCX	rw-	ZSoft IBM PC Paintbrush
PDB	PDB	rw+	Palm Database ImageViewer Format
PDF	PDF	rw+	Portable Document Format
PGM	PNM	rw+	Portable graymap format (gray scale)
PGX	PGX	r--	JPEG-2000 VM Format
PICON	XPM	rw-	Personal Icon
PICT	PICT	rw-	Apple Macintosh QuickDraw/PICT
PIX	PIX	r--	Alias/Wavefront RLE image format
PJPEG	JPEG	rw-	Progressive Joint Photographic Experts Group JFIF
PNG	PNG	rw-	Portable Network Graphics (libpng 1.2.8)
PNG24	PNG	rw-	24-bit RGB PNG, opaque only (zlib 1.2.3)
PNG32	PNG	rw-	32-bit RGBA PNG, semitransparency OK
PNG8	PNG	rw-	8-bit indexed PNG, binary transparency only
PNM	PNM	rw+	Portable anymap
PPM	PNM	rw+	Portable pixmap format (color)
PS	PS	rw+	PostScript
PSD	PSD	rw+	Adobe Photoshop bitmap
PTIF	TIFF	rw-	Pyramid encoded TIFF
PWP	PWP	r--	Seattle Film Works
R	RAW	rw+	Raw red samples
RAS	SUN	rw+	SUN Rasterfile
RGB	RGB	rw+	Raw red, green, and blue samples
RGBA	RGB	rw+	Raw red, green, blue, and alpha samples
RGBO	RGB	rw+	Raw red, green, blue, and opacity samples
RLA	RLA	r--	Alias/Wavefront image
RLE	RLE	r--	Utah Run length encoded image

File Extension	Coder	Read/ Write/ Multi-page	Format
SCR	SCR	r--	ZX-Spectrum SCREEN\$
SCT	SCT	r--	Scitex HandShake
SFW	SFW	r--	Seattle Film Works
SGI	SGI	rw+	Irix RGB image
STEGANO	STEGANO	r--	Steganographic image
SUN	SUN	rw+	SUN Rasterfile
SVG	SVG	rw+	Scalable Vector Graphics (XML 2.4.19)
SVGZ	SVG	rw+	Compressed Scalable Vector Graphics (XML 2.4.19)
TEXT	TXT	rw+	Text
TGA	TGA	rw+	Truevision Targa image
TIFF	TIFF	rw+	Tagged Image File Format (LIBTIFF, Version 3.8.2)
TIM	TIM	r--	PSX TIM
TXT	TXT	rw+	Text
UYVY	UYVY	rw-	16bit/pixel interleaved YUV
VDA	TGA	rw+	Truevision Targa image
VICAR	VICAR	rw-	VICAR rasterfile format
VID	VID	rw+	Visual Image Directory
VIFF	VIFF	rw+	Khoros Visualization image
VST	TGA	rw+	Truevision Targa image
WBMP	WBMP	rw-	Wireless Bitmap (level 0) image
WMF	WMF	r--	Windows Meta File
WMFWIN32	WMFWIN32	r--	Windows WIN32 API rendered Meta File
WMZ	WMZ	r--	Compressed Windows Meta File
WPG	WPG	r--	Word Perfect Graphics
X	X	rw+	X Image
XBM	XBM	rw-	X Windows system bitmap (black and white)
XC	XC	r--	Constant image uniform color
XCF	XCF	r--	GIMP image
XPM	XPM	rw-	X Windows system pixmap (color)
XV	VIFF	rw+	Khoros Visualization image
XWD	XWD	rw-	X Windows system window dump (color)
Y	RAW	rw+	Raw yellow samples
YCbCr	YCbCr	rw+	Raw Y, Cb, and Cr samples
YCbCrA	YCbCr	rw+	Raw Y, Cb, Cr, and opacity samples
YUV*	YUV	rw-	CCIR 601 4:1:1 or 4:2:2

where **r** indicates the format can be read (that is decoded), **w** indicates the format can be written (that is encoded) and **+** indicates the format handles multi-page images.

The default list of image formats supported by EMu is:

- BMP - Microsoft Windows Bitmap Image
- DCX - ZSoft IBM PC multi-page Paintbrush
- GIF - CompuServe Graphics Interchange Format
- JP2 - JPEG 2000
- JPEG - Joint Photographic Experts Group
- PCD - Photo CD
- PCX - ZSoft IBM PC Paintbrush
- PNG - Portable Network Graphics
- TGA - Truevision Targa Image
- TIFF - Tagged Image File Format

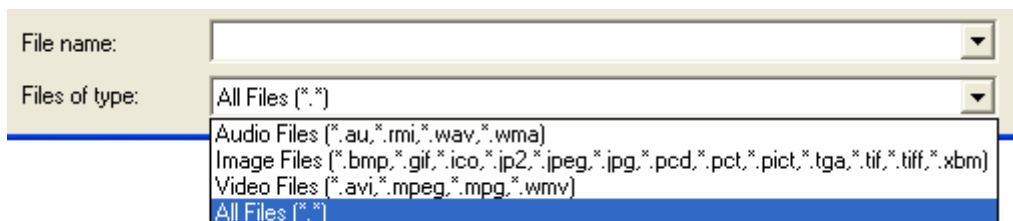
however, the list is typically longer, depending on the software installed on the computer. As with the audio and video media files, EMu provides a Registry entry for limiting the list of file extension shown when adding a file and generating a derivative. The format of the entry is:

```
User|user|Setting|Multimedia|Images|Exclude|format;...
Group|group|Setting|Multimedia|Images|Exclude|format;...
Group|Default|Setting|Multimedia|Images|Exclude|format;...
System|Setting|Multimedia|Images|Exclude|format;...
```

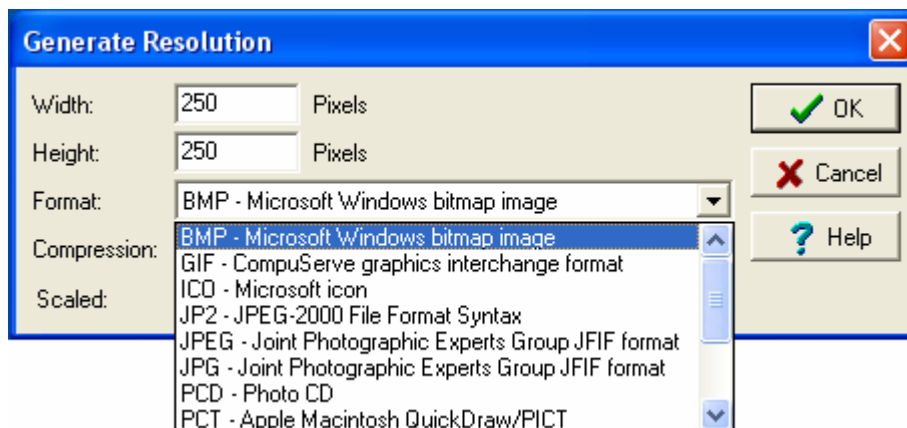
where *format* is one of the file extensions listed in the previous table. For example, if your institution does not support DCX, PNG and PCX as image formats, the following entry will remove them from the *Image Files* list and from the list of file types that can be generated:

```
System|Setting|Multimedia|Images|Exclude|dcx;png;pcx
```

The *Image Files* list would look similar to:



and the generate resolution dialogue would look similar to:



As with audio and video formats, image types excluded may be added to EMu. However EMu will not recognise them as an image format and will not display the image on the *Multimedia* tab. The file will be treated as a document, similar to say a Microsoft Project document.

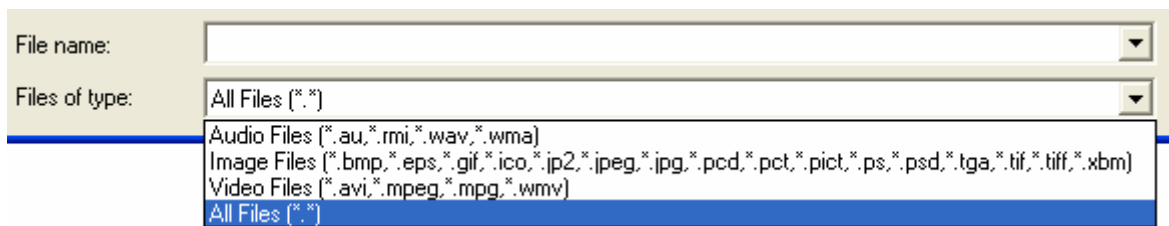
It is also possible to include image formats that are not added to the list by default. The following Registry entries can be used to add files extensions to the *Image Files* list:

```
User | user | Setting | Multimedia | Images | Include | format;...
Group | group | Setting | Multimedia | Images | Include | format;...
Group | Default | Setting | Multimedia | Images | Include | format;...
System | Setting | Multimedia | Images | Include | format;...
```

where *format* is a file extension to be included. For example, if group **Admin** is allowed to add PostScript and PhotoShop files, while other users were not, the following entry could be used:

```
Group | Admin | Multimedia | Images | Include | psd;eps;ps
```

For users in group **Admin** the *Image Files* list would look like:



Metadata

EMu 3.2.03 introduces support for three metadata standards:

- [EXIF](#) - *Exchangeable Image File Format* (Version 2.2, April 2002).
- [IPTC](#) - *International Press Telecommunications Council* (IIM Version 4.1, July 1999).
- [XMP](#) - *Extensible Metadata Platform* (Revision September 2005).

When image files are imported into the Multimedia module EMu examines the image file to determine whether metadata for any of the three supported standards is embedded. If metadata is found, it is extracted and used to populate the EXIF, IPTC and XMP tabs in the Multimedia module.

ImageMagick provides basic support for metadata extraction for a limited set of image types. The table below details the support provided:

Metadata Profile	Image Formats Supported
EXIF	JPEG, TIFF
IPTC	JPEG, TIFF, PICT, PS, PSD
XMP	JPEG, TIFF

It is expected that future releases of ImageMagick will provide support for metadata extraction from more image formats, in particular for XMP.

EMu provides a mechanism that allows institutions to embed pieces of metadata from the master image in derived images. For example, you may want to embed the copyright statement in all derivatives. It is also possible to have metadata values extracted into fields in the Multimedia module. Both these facilities are controlled by Registry entries (described below). First though it is worth looking at each of the metadata standards separately.

EXIF

EXIF or the Exchangeable Image File Format is a standard produced by the Japan Electronics and Information Technology Industries Association. The standard specifies the formats to be used for images, sound and tags in digital still cameras and in other systems handling the image and sound files recorded by digital still cameras. The data embedded generally relates to the physical attributes of the image (e.g. width, height, whether a flash was used, camera shutter speed, F stop used, etc.) rather than attributes about the subject of the image (e.g. keywords, location, etc.). As such, EXIF metadata is closely tied to the master image as it describes the physical attributes of that image.

EXIF information is embedded in an image using the TIFF 6.0 tags specification. Each EXIF property (that is a setting value, like Flash Fired - indicating whether the flash was used) is described by two numbers. The first number, called the IFD (Image File Directory) represents the broad area within which the property resides. EXIF provides five possible IFDs:

- 1 - [Primary Image IFD](#) (*Primary*)
- 2 - [EXIF specific IFD](#) (*Exif*)
- 3 - [GPS \(Global Positioning System\) IFD](#) (*Gps*)
- 4 - [Interoperability IFD](#) (*Interoperability*)
- 5 - [Thumbnail IFD](#) (*Thumbnail*)

Each IFD has its own set of numbers that represent specific attributes within that IFD; each number is known as a *tag*. For example, the EXIF IFD has a tag number 36867 that contains the date and time the picture was taken. By combining the IFD and the tag we get a unique identifier for a metadata value. The value 2:36867

represents the 2nd IFD (EXIF specific) with tag 36867 (date and time picture was taken). The **IFD:Tag** notation is used by EMu to specify EXIF metadata values.

The tables below describe the possible tags for each IFD. For a complete description of the values available for a given attribute, please consult the [EXIF Version 2.2 standard](#).

Primary Image IFD (IFD = 1)

Code	Name	Short description
254	NewSubfileType	A general indication of the kind of data contained in this subfile.
255	SubfileType	A general indication of the kind of data contained in this subfile.
256	ImageWidth	The number of columns in the image, i.e., the number of pixels per row.
257	ImageLength	The number of rows of pixels in the image.
258	BitsPerSample	Number of bits per component.
259	Compression	Compression scheme used on the image data.
262	PhotometricInterpretation	The color space of the image data.
263	Threshholding	For black and white TIFF files that represent shades of gray, the technique used to convert from gray to black and white pixels.
264	CellWidth	The width of the dithering or halftoning matrix used to create a dithered or halftoned bilevel file.
265	CellLength	The length of the dithering or halftoning matrix used to create a dithered or halftoned bilevel file.
266	FillOrder	The logical order of bits within a byte.
270	ImageDescription	A string that describes the subject of the image.
271	Make	The scanner manufacturer.
272	Model	The scanner model name or number.
273	StripOffsets	For each strip, the byte offset of that strip.
274	Orientation	The orientation of the image with respect to the rows and columns.
277	SamplesPerPixel	The number of components per pixel.
278	RowsPerStrip	The number of rows per strip.
279	StripByteCounts	For each strip, the number of bytes in the strip after compression.
280	MinSampleValue	The minimum component value used.
281	MaxSampleValue	The maximum component value used.
282	XResolution	The number of pixels per ResolutionUnit in the ImageWidth direction.
283	YResolution	The number of pixels per ResolutionUnit in the ImageLength direction.
284	PlanarConfiguration	How the components of each pixel are stored.
288	FreeOffsets	For each string of contiguous unused bytes in a TIFF file, the byte offset of the string.
289	FreeByteCounts	For each string of contiguous unused bytes in a TIFF file, the number of bytes in the string.
290	GrayResponseUnit	The precision of the information contained in the GrayResponseCurve.
291	GrayResponseCurve	For grayscale data, the optical density of each possible pixel value.
296	ResolutionUnit	The unit of measurement for XResolution and YResolution.
305	Software	Name and version number of the software package(s) used to create the

Code	Name	Short description
		image.
306	DateTime	Date and time of image creation.
315	Artist	Person who created the image.
316	HostComputer	The computer and/or operating system in use at the time of image creation.
320	ColorMap	A color map for palette color images.
338	ExtraSamples	Description of extra components.
33432	Copyright	Copyright notice.

EXIF IFD (IFD = 2)

Code	Name	Short description
33434	ExposureTime	Exposure time, given in seconds.
33437	FNumber	The F number.
34850	ExposureProgram	The class of the program used by the camera to set exposure when the picture is taken.
34852	SpectralSensitivity	Indicates the spectral sensitivity of each channel of the camera used.
34855	ISOSpeedRatings	Indicates the ISO Speed and ISO Latitude of the camera or input device as specified in ISO 12232.
34856	OECF	Indicates the Opto-Electric Conversion Function (OECF) specified in ISO 14524.
36864	ExifVersion	The version of the supported Exif standard.
36867	DateTimeOriginal	The date and time when the original image data was generated.
36868	DateTimeDigitized	The date and time when the image was stored as digital data.
37121	ComponentsConfiguration	Specific to compressed data; specifies the channels and complements PhotometricInterpretation
37122	CompressedBitsPerPixel	Specific to compressed data; states the compressed bits per pixel.
37377	ShutterSpeedValue	Shutter speed.
37378	ApertureValue	The lens aperture.
37379	BrightnessValue	The value of brightness.
37380	ExposureBiasValue	The exposure bias.
37381	MaxApertureValue	The smallest F number of the lens.
37382	SubjectDistance	The distance to the subject, given in meters.
37383	MeteringMode	The metering mode.
37384	LightSource	The kind of light source.
37385	Flash	Indicates the status of flash when the image was shot.
37386	FocalLength	The actual focal length of the lens, in mm.
37396	SubjectArea	Indicates the location and area of the main subject in the overall scene.
37500	MakerNote	Manufacturer specific information.
37510	UserComment	Keywords or comments on the image; complements ImageDescription.
37520	SubsecTime	A tag used to record fractions of seconds for the DateTime tag.

Code	Name	Short description
37521	SubsecTimeOriginal	A tag used to record fractions of seconds for the DateTimeOriginal tag.
37522	SubsecTimeDigitized	A tag used to record fractions of seconds for the DateTimeDigitized tag.
40960	FlashpixVersion	The Flashpix format version supported by a FPXR file.
40961	ColorSpace	The color space information tag is always recorded as the color space specifier.
40962	PixelXDimension	Specific to compressed data; the valid width of the meaningful image.
40963	PixelYDimension	Specific to compressed data; the valid height of the meaningful image.
40964	RelatedSoundFile	Used to record the name of an audio file related to the image data.
41483	FlashEnergy	Indicates the strobe energy at the time the image is captured, as measured in Beam Candle Power Seconds
41484	SpatialFrequencyResponse	Records the camera or input device spatial frequency table and SFR values in the direction of image width, image height, and diagonal direction, as specified in ISO 12233.
41486	FocalPlaneXResolution	Indicates the number of pixels in the image width (X) direction per FocalPlaneResolutionUnit on the camera focal plane.
41487	FocalPlaneYResolution	Indicates the number of pixels in the image height (Y) direction per FocalPlaneResolutionUnit on the camera focal plane.
41488	FocalPlaneResolutionUnit	Indicates the unit for measuring FocalPlaneXResolution and FocalPlaneYResolution.
41492	SubjectLocation	Indicates the location of the main subject in the scene.
41493	ExposureIndex	Indicates the exposure index selected on the camera or input device at the time the image is captured.
41495	SensingMethod	Indicates the image sensor type on the camera or input device.
41728	FileSource	Indicates the image source.
41729	SceneType	Indicates the type of scene.
41730	CFAPattern	Indicates the color filter array (CFA) geometric pattern of the image sensor when a one-chip color area sensor is used.
41985	CustomRendered	Indicates the use of special processing on image data, such as rendering geared to output.
41986	ExposureMode	Indicates the exposure mode set when the image was shot.
41987	WhiteBalance	Indicates the white balance mode set when the image was shot.
41988	DigitalZoomRatio	Indicates the digital zoom ratio when the image was shot.
41989	FocalLengthIn35mmFilm	Indicates the equivalent focal length assuming a 35mm film camera, in mm.
41990	SceneCaptureType	Indicates the type of scene that was shot.
41991	GainControl	Indicates the degree of overall image gain adjustment.
41992	Contrast	Indicates the direction of contrast processing applied by the camera when the image was shot.
41993	Saturation	Indicates the direction of saturation processing applied by the camera when the image was shot.
41994	Sharpness	Indicates the direction of sharpness processing applied by the camera when the image was shot.
41995	DeviceSettingDescription	This tag indicates information on the picture-taking conditions of a

Code	Name	Short description
		particular camera model.
41996	SubjectDistanceRange	Indicates the distance to the subject.
42016	ImageUniqueID	Indicates an identifier assigned uniquely to each image.

GPS IFD (IFD = 3)

Code	Name	Short description
0	GPSVersionID	Indicates the version of GPSInfoIFD.
1	GPSLatitudeRef	Indicates whether the latitude is north or south latitude.
2	GPSLatitude	Indicates the latitude.
3	GPSLongitudeRef	Indicates whether the longitude is east or west longitude.
4	GPSLongitude	Indicates the longitude.
5	GPSAltitudeRef	Indicates the altitude used as the reference altitude.
6	GPSAltitude	Indicates the altitude based on the reference in GPSAltitudeRef.
7	GPSTimeStamp	Indicates the time as UTC (Coordinated Universal Time).
8	GPSSatellites	Indicates the GPS satellites used for measurements.
9	GPSStatus	Indicates the status of the GPS receiver when the image is recorded.
10	GPSMeasureMode	Indicates the GPS measurement mode.
11	GPSDOP	Indicates the GPS DOP (data degree of precision).
12	GPSSpeedRef	Indicates the unit used to express the GPS receiver speed of movement.
13	GPSSpeed	Indicates the speed of GPS receiver movement.
14	GPSTrackRef	Indicates the reference for giving the direction of GPS receiver movement.
15	GPSTrack	Indicates the direction of GPS receiver movement.
16	GPSImgDirectionRef	Indicates the reference for giving the direction of the image when it is captured.
17	GPSImgDirection	Indicates the direction of the image when it was captured.
18	GPSMapDatum	Indicates the geodetic survey data used by the GPS receiver.
19	GPSDestLatitudeRef	Indicates whether the latitude of the destination point is north or south latitude.
20	GPSDestLatitude	Indicates the latitude of the destination point.
21	GPSDestLongitudeRef	Indicates whether the longitude of the destination point is east or west longitude.
22	GPSDestLongitude	Indicates the longitude of the destination point.
23	GPSDestBearingRef	Indicates the reference used for giving the bearing to the destination point.
24	GPSDestBearing	Indicates the bearing to the destination point.
25	GPSDestDistanceRef	Indicates the unit used to express the distance to the destination point.
26	GPSDestDistance	Indicates the distance to the destination point.
27	GPSProcessingMethod	A character string recording the name of the method used for location finding.
28	GPSAreaInformation	A character string recording the name of the GPS area.
29	GPSDateStamp	A character string recording date and time information relative to UTC

Code	Name	Short description
		(Coordinated Universal Time).
30	GPSDifferential	Indicates whether differential correction is applied to the GPS receiver.

Interoperability IFD (IFD = 4)

Code	Name	Short description
1	InteroperabilityIndex	Indicates the identification of the Interoperability rule.

Thumbnail IFD (IFD = 5)

The codes used by the thumbnail IFD are exactly the same as for the [Primary Image IFD](#).

IPTC

The International Press Telecommunications Council released a metadata standard in 1990 that was to allow contextual information to be embedded within press photographs. The standard became known as the IPTC standard. IPTC focuses on information about the subject of the image rather than its physical attributes, thus it contains information such as keywords, copyright statements, location details and subject of the image. In general, the IPTC information is as relevant to any image derivative as to the master image.

IPTC metadata is stored in a similar format to that used for EXIF. IPTC has the concept of a broad category of attributes, known as a **Record**. Within a record a series of tags is used to specify individual attributes. IPTC defines nine **Record** values, however only 6 are used by images:

- 1 - [Envelope](#) (*Envelope*)
- 2 - [Application](#) (*Application*)
- 3 - [News Photo](#) (*NewsPhoto*)
- 7 - [Pre Object Data](#) (*PreObjectData*)
- 8 - [Object Data](#) (*ObjectData*)
- 9 - [Post Object Data](#) (*PostObjectData*)

As with EXIF the combination of a Record and Tag defines an individual attribute. The **Record:Tag** notation is used. Unlike EXIF however, IPTC does allow for repeat attributes, so attribute 2:25 (which is the keywords tag) may appear multiple times if more than one keyword is defined.

The tables below describe what tags are available on a **Record** basis. For a complete description of the values available for a given attribute, please consult the [IPTC IMM Specification](#).

Envelope (Record = 1)

Tag ID	Tag Name	Values / Notes
0	EnvelopeRecordVersion	
5	Destination	
20	FileFormat	0 = No ObjectData 1 = IPTC-NAA Digital Newsphoto Parameter Record 2 = IPTC7901 Recommended Message Format 3 = Tagged Image File Format (Adobe/Aldus Image data) 4 = Illustrator (Adobe Graphics data) 5 = AppleSingle (Apple Computer Inc) 6 = NAA 89-3 (ANPA 1312) 7 = MacBinary II

Tag ID	Tag Name	Values / Notes
		8 = IPTC Unstructured Character Oriented File Format (UCOFF) 9 = United Press International ANPA 1312 variant 10 = United Press International Down-Load Message 11 = JPEG File Interchange (JFIF) 12 = Photo-CD Image-Pac (Eastman Kodak) 13 = Bit Mapped Graphics File [.BMP] (Microsoft) 14 = Digital Audio File [.WAV] (Microsoft & Creative Labs) 15 = Audio plus Moving Video [.AVI] (Microsoft) 16 = PC DOS/Windows Executable Files [.COM][.EXE] 17 = Compressed Binary File [.ZIP] (PKWare Inc) 18 = Audio Interchange File Format AIFF (Apple Computer Inc) 19 = RIFF Wave (Microsoft Corporation) 20 = Freehand (Macromedia/Aldus) 21 = Hypertext Markup Language [.HTML] (The Internet Society) 22 = MPEG 2 Audio Layer 2 (Musicom), ISO/IEC 23 = MPEG 2 Audio Layer 3, ISO/IEC 24 = Portable Document File [.PDF] Adobe 25 = News Industry Text Format (NITF) 26 = Tape Archive [.TAR] 27 = Tidningarnas Telegrambyra NITF version (TTNITF DTD) 28 = Ritzaus Bureau NITF version (RBNITF DTD) 29 = Corel Draw [.CDR]
22	FileVersion	
30	ServiceIdentifier	
40	EnvelopeNumber	
50	ProductID	
60	EnvelopePriority	
70	DateSent	
80	TimeSent	
90	CodedCharacterSet	(values are entered in the form "ESC X Y[, ...]". The escape sequence for UTF-8 character coding is "ESC % G", but this is displayed as "UTF8" for convenience. Either string may be used when writing. The value of this tag affects the decoding of string values in the Application and NewsPhoto records)
100	UniqueObjectName	
120	ARMIdentifier	
122	ARMVersion	

Application (Record = 2)

Tag ID	Tag Name	Values / Notes
0	ApplicationRecordVersion	
3	ObjectTypeReference	
4	ObjectAttributeReference	
5	ObjectName	
7	EditStatus	
8	EditorialUpdate	

Tag ID	Tag Name	Values / Notes
10	Urgency	
12	SubjectReference	
15	Category	
20	SupplementalCategories	
22	FixtureIdentifier	
25	Keywords	
26	ContentLocationCode	
27	ContentLocationName	
30	ReleaseDate	
35	ReleaseTime	
37	ExpirationDate	
38	ExpirationTime	
40	SpecialInstructions	
42	ActionAdvised	01 = Object Kill 02 = Object Replace 03 = Object Append 04 = Object Reference
45	ReferenceService	
47	ReferenceDate	
50	ReferenceNumber	
55	DateCreated	
60	TimeCreated	
62	DigitalCreationDate	
63	DigitalCreationTime	
65	OriginatingProgram	
70	ProgramVersion	
75	ObjectCycle	'a' = Morning 'b' = Both Morning and Evening 'p' = Evening
80	By-line	
85	By-lineTitle	
90	City	
92	Sub-location	
95	Province-State	
100	Country-PrimaryLocationCode	
101	Country-PrimaryLocationName	
103	OriginalTransmissionReference	
105	Headline	
110	Credit	
115	Source	

Tag ID	Tag Name	Values / Notes
116	CopyrightNotice	
118	Contact	
120	Caption-Abstract	
121	LocalCaption	
122	Writer-Editor	
125	RasterizedCaption	
130	ImageType	
131	ImageOrientation	'L' = Landscape 'P' = Portrait 'S' = Square
135	LanguageIdentifier	
150	AudioType	'0T' = Text Only '1A' = Mono Actuality '1C' = Mono Question and Answer Session '1M' = Mono Music '1Q' = Mono Response to a Question '1R' = Mono Raw Sound '1S' = Mono Scener '1V' = Mono Voicer '1W' = Mono Wrap '2A' = Stereo Actuality '2C' = Stereo Question and Answer Session '2M' = Stereo Music '2Q' = Stereo Response to a Question '2R' = Stereo Raw Sound '2S' = Stereo Scener '2V' = Stereo Voicer '2W' = Stereo Wrap
151	AudioSamplingRate	
152	AudioSamplingResolution	
153	AudioDuration	
154	AudioOutcue	
184	JobID	
185	MasterDocumentID	
186	ShortDocumentID	
187	UniqueDocumentID	
188	OwnerID	
200	ObjectPreviewFileFormat	0 = No ObjectData 1 = IPTC-NAA Digital Newsphoto Parameter Record 2 = IPTC7901 Recommended Message Format 3 = Tagged Image File Format (Adobe/Aldus Image data) 4 = Illustrator (Adobe Graphics data) 5 = AppleSingle (Apple Computer Inc) 6 = NAA 89-3 (ANPA 1312) 7 = MacBinary II 8 = IPTC Unstructured Character Oriented File Format (UCOFF) 9 = United Press International ANPA 1312 variant

Tag ID	Tag Name	Values / Notes
		10 = United Press International Down-Load Message 11 = JPEG File Interchange (JFIF) 12 = Photo-CD Image-Pac (Eastman Kodak) 13 = Bit Mapped Graphics File [.BMP] (Microsoft) 14 = Digital Audio File [.WAV] (Microsoft & Creative Labs) 15 = Audio plus Moving Video [.AVI] (Microsoft) 16 = PC DOS/Windows Executable Files [.COM][.EXE] 17 = Compressed Binary File [.ZIP] (PKWare Inc) 18 = Audio Interchange File Format AIFF (Apple Computer Inc) 19 = RIFF Wave (Microsoft Corporation) 20 = Freehand (Macromedia/Aldus) 21 = Hypertext Markup Language [.HTML] (The Internet Society) 22 = MPEG 2 Audio Layer 2 (Musicom), ISO/IEC 23 = MPEG 2 Audio Layer 3, ISO/IEC 24 = Portable Document File [.PDF] Adobe 25 = News Industry Text Format (NITF) 26 = Tape Archive [.TAR] 27 = Tidningarnas Telegrambyra NITF version (TTNITF DTD) 28 = Ritzaus Bureau NITF version (RBNITF DTD) 29 = Corel Draw [.CDR]
201	ObjectPreviewFileVersion	
202	ObjectPreviewData	
225	ClassifyState	
228	SimilarityIndex	
230	DocumentNotes	
231	DocumentHistory	
232	ExifCameraInfo	

News Photo (Record = 3)

Tag ID	Tag Name	Values / Notes
0	NewsPhotoVersion	
10	IPTCPictureNumber	(4 numbers: 1-Manufacturer ID, 2-Equipment ID, 3-Date, 4-Sequence)
20	IPTCImageWidth	
30	IPTCImageHeight	
40	IPTCPixelWidth	
50	IPTCPixelHeight	
55	SupplementalType	0 = Main Image 1 = Reduced Resolution Image 2 = Logo 3 = Rasterized Caption
60	ColorRepresentation	0x0 = No Image, Single Frame 0x100 = Monochrome, Single Frame 0x300 = 3 Components, Single Frame 0x301 = 3 Components, Frame Sequential in Multiple Objects 0x302 = 3 Components, Frame Sequential in One Object

Tag ID	Tag Name	Values / Notes
		0x303 = 3 Components, Line Sequential 0x304 = 3 Components, Pixel Sequential 0x305 = 3 Components, Special Interleaving 0x400 = 4 Components, Single Frame 0x401 = 4 Components, Frame Sequential in Multiple Objects 0x402 = 4 Components, Frame Sequential in One Object 0x403 = 4 Components, Line Sequential 0x404 = 4 Components, Pixel Sequential 0x405 = 4 Components, Special Interleaving
64	InterchangeColorSpace	1 = X,Y,Z CIE 2 = RGB SMPTE 3 = Y,U,V (K) (D65) 4 = RGB Device Dependent 5 = CMY (K) Device Dependent 6 = Lab (K) CIE 7 = YCbCr 8 = sRGB
65	ColorSequence	
66	ICC_Profile	
70	ColorCalibrationMatrix	
80	LookupTable	
84	NumIndexEntries	
85	ColorPalette	
86	IPTCBitsPerSample	
90	SampleStructure	0 = OrthogonalConstangSampling 1 = Orthogonal4-2-2Sampling 2 = CompressionDependent
100	ScanningDirection	0 = L-R, Top-Bottom 1 = R-L, Top-Bottom 2 = L-R, Bottom-Top 3 = R-L, Bottom-Top 4 = Top-Bottom, L-R 5 = Bottom-Top, L-R 6 = Top-Bottom, R-L 7 = Bottom-Top, R-L
102	IPTCImageRotation	0 = 0 1 = 90 2 = 180 3 = 270
110	DataCompressionMethod	
120	QuantizationMethod	0 = Linear Reflectance/Transmittance 1 = Linear Density 2 = IPTC Ref B 3 = Linear Dot Percent 4 = AP Domestic Analogue 5 = Compression Method Specific 6 = Color Space Specific 7 = Gamma Compensated
125	EndPoints	

Tag ID	Tag Name	Values / Notes
130	ExcursionTolerance	0 = Not Allowed 1 = Allowed
135	BitsPerComponent	
140	MaximumDensityRange	
145	GammaCompensatedValue	

Pre Object Data (Record = 7)

Tag ID	Tag Name	Values / Notes
10	SizeMode	0 = Size Not Known 1 = Size Known
20	MaxSubfileSize	
90	ObjectSizeAnnounced	
95	MaximumObjectSize	

Object Data (Record = 8)

Tag ID	Tag Name	Values / Notes
10	SubFile	

Post Object Data (Record = 9)

Tag ID	Tag Name	Values / Notes
10	ConfirmedObjectSize	

XMP

In April 2001 Adobe released the **Extensible Metadata Platform** or **XMP** standard. Unlike EXIF and IPTC, which are used to describe specific image attributes, XMP is a standard that defines how other metadata standard (like EXIF, IPTC, Dublin Core, etc.) should be embedded. It is essentially a wrapper around other metadata items. It defines how the metadata items should be embedded and how they should be stored. It uses XML to represent values rather than tags as used by EXIF and IPTC.

XMP also uses RDF (Resource Description Framework). RDF is a set of XML tags that allows data attributes to be defined. These attributes can be extracted from images (and other document types) by search engines to allow users to locate documents based on embedded metadata. Using RDF makes it possible to search for documents of any format rather than just HTML pages. This is the basis of the *semantic web*.

Since XMP does not define attributes, but rather uses other metadata standards there is no set of “tags” specific to XMP. Note that the other metadata standards do define attributes. So XMP provides a simple wrapper around other metadata types. A sample XMP document looks like:

```
<?xpacket begin="~K»¿" id="W5M0MpCehiHzreSzNTczkc9d"?>
<x:xmpmeta xmlns:x="adobe:ns:meta/" x:xmptk="3.1.1-111">
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
<rdf:Description rdf:about=""
xmlns:tiff="http://ns.adobe.com/tiff/1.0/"
xmlns:xap="http://ns.adobe.com/xap/1.0/"
xmlns:exif="http://ns.adobe.com/exif/1.0/"
xmlns:photoshop="http://ns.adobe.com/photoshop/1.0/"
xmlns:Iptc4xmpCore="http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/"
```



```

xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:xapRights="http://ns.adobe.com/xap/1.0/rights/"
tiff:Orientation="1"
tiff:YCbCrPositioning="2"
tiff:XResolution="144/1"
tiff:YResolution="144/1"
tiff:ResolutionUnit="2"
tiff:Make="OLYMPUS OPTICAL CO.,LTD"
tiff:Model="E-20,E-20N,E-20P"
xap:CreatorTool="29-1104"
xap:MetadataDate="2005-09-07T15:16:34+10:00"
exif:ExifVersion="0.2.1.0"
exif:FlashpixVersion="0.1.0.0"
exif:ColorSpace="1"
exif:PixelXDimension="2560"
exif:PixelYDimension="1920"
exif:DateTimeOriginal="2005-05-11T14:19:42+10:00"
exif:DateTimeDigitized="2005-05-11T14:19:42+10:00"
exif:ExposureTime="1/400"
exif:FNumber="40/10"
photoshop:ICCProfile="sRGB IEC61966-2.1"
photoshop:ColorMode="3"
photoshop:City="Broome"
photoshop:State="WA"
photoshop:Country="Australia"
Iptc4xmpCore:Location="Broome environs">
<tiff:BitsPerSample>
  <rdf:Seq>
    <rdf:li>8</rdf:li>
    <rdf:li>8</rdf:li>
    <rdf:li>8</rdf:li>
  </rdf:Seq>
</tiff:BitsPerSample>
<exif:ComponentsConfiguration>
  <rdf:Seq>
    <rdf:li>1</rdf:li>
    <rdf:li>2</rdf:li>
    <rdf:li>3</rdf:li>
    <rdf:li>0</rdf:li>
  </rdf:Seq>
</exif:ComponentsConfiguration>
<exif:ISOSpeedRatings>
  <rdf:Seq>
    <rdf:li>80</rdf:li>
  </rdf:Seq>
</exif:ISOSpeedRatings>
<exif:Flash
  exif:Fired="False"
  exif:Return="0"
  exif:Mode="0"
  exif:Function="False"
  exif:RedEyeMode="False"/>
<dc:description>
  <rdf:Alt>
    <rdf:li xml:lang="x-default">Strapping camera to Holden for car journey
scene.</rdf:li>
  </rdf:Alt>
</dc:description>
<dc:rights>
  <rdf:Alt>
    <rdf:li xml:lang="x-default">Copyright, all rights reserved</rdf:li>
  </rdf:Alt>
</dc:rights>
<dc:title>

```

```

<rdf:Alt>
  <rdf:li xml:lang="x-default">Holden street scene</rdf:li>
</rdf:Alt>
</dc:title>
<xapRights:UsageTerms>
  <rdf:Alt>
    <rdf:li xml:lang="x-default">Copyrighted Material, no reproduction without
prior permission</rdf:li>
  </rdf:Alt>
</xapRights:UsageTerms>
</rdf:Description>
</rdf:RDF>
</x:xmpmeta>

```

While you may not be able to understand XML a brief glance will show that EXIF (*exif:*), IPTC (*Iptc4xmpCore:*), TIFF (*tiff:*), Dublin Core (*dc:*) and other metadata standards are embedded. The RDF (*rdf:*) tags define the XMP structure of the document and are not metadata attribute tags.

As XMP is XML-based it is extensible, meaning that new metadata standards can be embedded as they emerge without changing the XMP specification. It is also possible to define your own XML schema (set of tags for your use) and embed them using XMP.

There are two parts to the definition of a metadata attribute in XMP. The first part is the **namespace**. This is a label used to define a certain metadata standard. The second part is the XML **tag** within that namespace. To determine the **namespace:tag** pair for a metadata attribute you must first locate the tag in the XML. In the above XML let's use `<dc:title>` as our example. The **tag** part of this XML is **title**. The **dc:** on the front is known as the **prefix**. To determine the **namespace** we need to use the prefix to locate the namespace value. Namespaces have a tag of `xmlns:prefix="namespace"`. So you may be able to locate the line:

```
xmlns:dc="http://purl.org/dc/elements/1.1/"
```

This entry defines the namespace for the **dc** prefix to be `http://purl/dc/elements/1.1/`. Thus to define the title element we need to use the form `http://purl/dc/elements/1.1/:title`. The reason why the prefix cannot be used is because XML allows any prefix value to be defined for a given namespace (so **dc** does not have to be the prefix).

Any namespace can be used in XMP, however there are some standards-based predefined namespaces. The table below details them. For a complete description of the XMP standard, please consult the [XMP specification](#).

Namespace	Metadata Standard
http://purl.org/dc/elements/1.1/	Dublin Core Schema
http://ns.adobe.com/xap/1.0/	XMP Basic Schema
http://ns.adobe.com/xap/1.0/rights/	XMP Rights Management Schema
http://ns.adobe.com/xap/1.0/mm/	XMP Media Management Schema
http://ns.adobe.com/xap/1.0/bj/	XMP Basic Job Ticket Schema
http://ns.adobe.com/xap/1.0/t/pg/	XMP Paged-Text Schema
http://ns.adobe.com/xmp/1.0/DynamicMedia/	XMP Paged-Text Schema
http://ns.adobe.com/pdf/1.3/	Adobe PDF Schema
http://ns.adobe.com/photoshop/1.0/	Photoshop Schema
http://ns.adobe.com/camera-rawsettings/1.0/	Camera Raw Schema
http://ns.adobe.com/tiff/1.0/	EXIF Schema for TIFF Properties

Namespace	Metadata Standard
http://ns.adobe.com/exif/1.0/	EXIF Schema for EXIF-specific Properties
http://ns.adobe.com/exif/1.0/aux/	EXIF Schema for Additional EXIF Properties
http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/	IPTC Core Schema

Embed in derivatives

When derivatives are generated in EMu (via the *Multimedia->Generate Resolution* command) a Registry entry is used to determine what metadata from the master image should be propagated to the derived image. The entry allows individual EXIF, IPTC and XMP properties to be selected or a whole section may be selected. The format of the entry is:

```
User | user | Setting | Multimedia | Metadata | Embed | type | format | tag; ...
Group | group | Setting | Multimedia | Metadata | Embed | type | format | tag; ...
Group | Default | Setting | Multimedia | Metadata | Embed | type | format | tag; ...
System | Setting | Multimedia | Metadata | Embed | type | format | tag; ...
```

where *type* is one of *EXIF*, *IPTC* or *XMP*. The *format* defines the image format of the derivative for which the entry applies. There are three alternatives for this value:

Default

The metadata tags defined in the *tag* section apply for all derivatives generated.

format

An image format supported by ImageMagick. For example a value of **JPEG** would mean the metadata is only embedded for JPEG derivatives.

width:height:format

If the derivative matches the supplied *width* and *height* and is for the given image format, the metadata is embedded. For example, a value of **90:90:JPEG** would indicate that the entry applies when generating a 90 pixel by 90 pixel JPEG image.

When determining which Registry entries apply for embedding tags in derivatives, EMu first looks for the third entry type (that is the most specific entry); if one is not found, it looks for the second entry type; otherwise the first. If an entry is not found, no tags are embedded in the derivative.

The *tag* value depends on the type of metadata being embedded. The allowable values are:

All

All the metadata for the *type* (that is EXIF, IPTC or XMP) is embedded in the derivative.

None

None of the metadata for the *type* is embedded in the derivative.

attribute

The attribute value to embed in the derivative. The format of the attribute depends on the type of metadata being embedded. For EXIF data it is an *IFD:tag* pair; for IPTC it is a *Record:tag* pair; and for XMP it is a *namespace:tag* pair. If the *tag* part of the pair is left empty, all attributes for that grouping (IFD, Record or namespace) are included. An *attribute* may also have a preceding minus sign (-) implying the attribute should not be embedded.

In this first example the EXIF copyright property (attribute 1:33432) is to be embedded in all derived images.

```
System | Setting | Multimedia | Metadata | Embed | EXIF | Default | 1:33432
```

If we want the EXIF copyright text copied into all images except for thumbnails (90 pixel by 90 pixel JPEG), the following two entries would be required:

```
System|Setting|Multimedia|Metadata|Embed|EXIF|Default|1:33432
System|Setting|Multimedia|Metadata|Embed|EXIF|90:90:JPEG|None
```

The next entries will embed the XMP Dublin Core XML in derived TIFF and JPEG images:

```
System|Setting|Multimedia|Metadata|Embed|XMP|TIFF|http://purl.org/dc/elements/1.1/:
System|Setting|Multimedia|Metadata|Embed|XMP|JPEG|http://purl.org/dc/elements/1.1/:
```

Notice how only the namespace is specified (it is important to have the trailing colon), indicating that all Dublin Core attributes should be embedded.

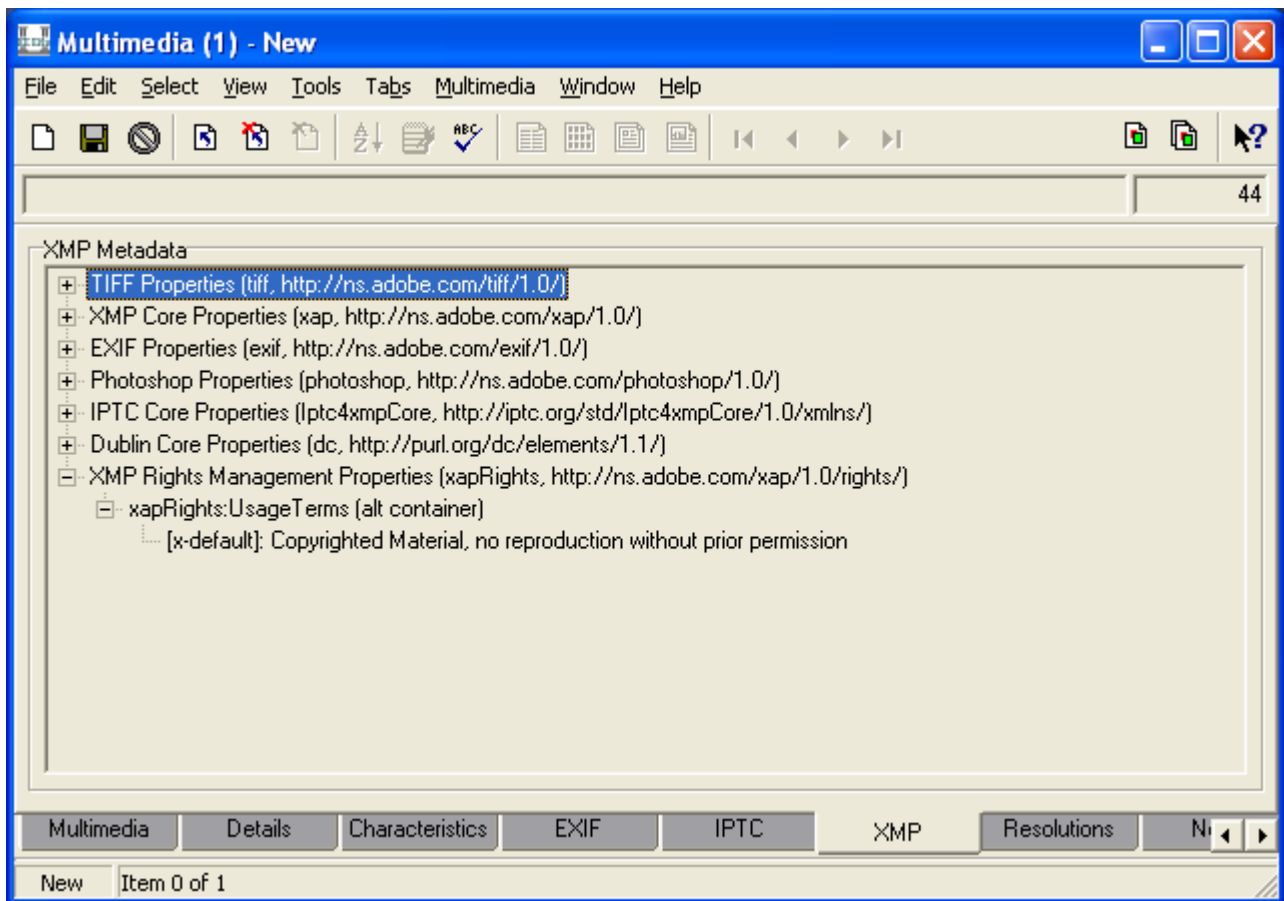
The next entry will embed all IPTC Application attributes (Record type 2) except for the *MasterDocumentID* and the *UniqueDocumentID* in all derivatives:

```
System|Setting|Multimedia|Metadata|Embed|IPTC|Default|2::-2:185;-2:187
```

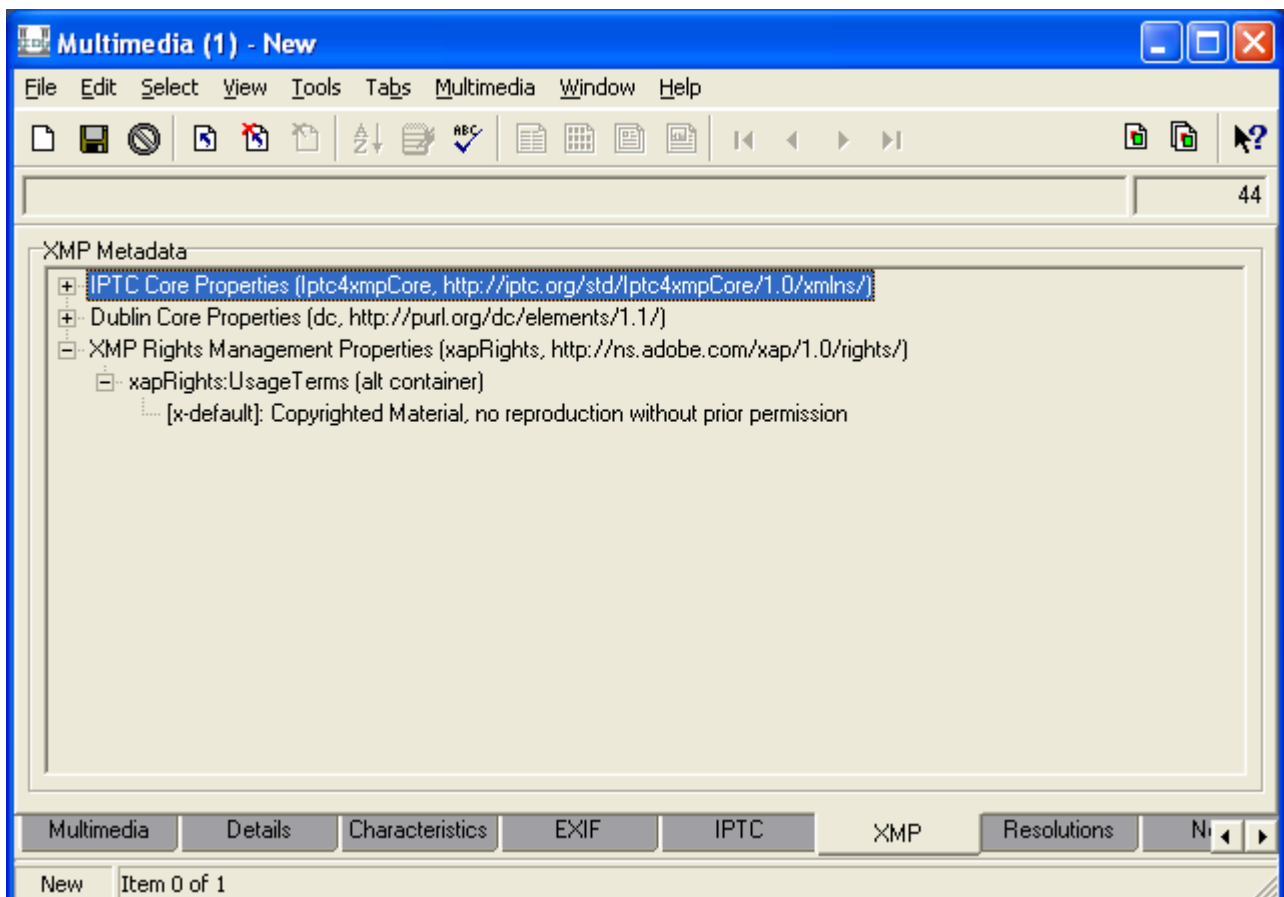
Finally the following list of entries will ensure that thumbnail images do not store any metadata. For all other derivatives all IPTC metadata is embedded and for XMP, Dublin Core, IPTC and Rights data is embedded:

```
System|Setting|Multimedia|Metadata|Embed|EXIF|Default|None
System|Setting|Multimedia|Metadata|Embed|IPTC|Default|All
System|Setting|Multimedia|Metadata|Embed|IPTC|90:90:JPEG|None
System|Setting|Multimedia|Metadata|Embed|XMP|Default|http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/:;http://ns.adobe.com/xap/1.0/rights/:;http://purl.org/dc/elements/1.1/:
System|Setting|Multimedia|Metadata|Embed|XMP|90:90:JPEG|None
```

If we assume the settings above and the XMP for the master image is displayed as follows:



then the derivative metadata would look like:



The ImageMagick libraries only support the embedding of EXIF information in JPEG images. They do not

provide support for TIFF. EMu does however provide basic support for embedding the following Primary (IFD = 1) attributes:

- Comment (1:270)
- Make (1:271)
- Model (1:272)
- Label (1:297)
- Timestamp (1:306)
- Artist (1:315)
- HostComputer (1:316)
- Copyright (1:33432)

Extract into Multimedia module

EMu allows metadata to be extracted and placed into fields in the Multimedia module. Support for both atomic values and for tables of values is provided. Data from all supported metadata types (EXIF, IPTC and XMP) may be extracted. The format of the Registry entry is:

```
User|user|Setting|Multimedia|Metadata|Extract|type|colname|tag;...
Group|group|Setting|Multimedia|Metadata|Extract|type|colname|tag;...
Group|Default|Setting|Multimedia|Metadata|Extract|type|colname|tag;...
System|Setting|Multimedia|Metadata|Extract|type|colname|tag;...
```

where *type* is one of *EXIF*, *IPTC* or *XMP*. The name of the column into which the metadata is to be placed is specified by *colname*. The *tag* value is an *IFD:tag* pair for EXIF, a *Record:tag* pair for IPTC or a *namespace:tag* pair for XMP. When a list of tags is supplied the first tag that has a value is used.

The metadata extracted will only be inserted into the specified *colname* if there is no data already present in the field. It will not overwrite or add to existing data. When extracting metadata that has multiple values into an atomic field, EMu will join the values, separating them by a comma. When data is placed into a table column, values are broken on newline characters. Where an attribute is repeated (e.g. IPTC keywords attribute) each entry is placed in a separate row in the table.

The following entry will extract the IPTC caption attribute (2:120) and add it to the *Title* field on the *Multimedia* tab:

```
System|Setting|Multimedia|Metadata|Extract|IPTC|MultiTitle|2:120
```

The next entry will extract the Dublin Core title attribute and place it in the *Title* field:

```
System|Setting|Multimedia|Metadata|Extract|XMP|MultiTitle|http://purl.org/dc/elements/1.1/:title
```

The next entry will extract the keywords from the IPTC metadata (2:25) and place them in the *Subject/Keywords* list on the *Details* tab:

```
System|Setting|Multimedia|Metadata|Extract|IPTC|DetSubject_tab|2:25
```

Finally the following three entries will try and extract the copyright details from EXIF, IPTC and XMP metadata and place them in the *Rights* field on the *Details* tab:

```
System|Setting|Multimedia|Metadata|Extract|EXIF|DetRights|1:33432
System|Setting|Multimedia|Metadata|Extract|IPTC|DetRights|2:116
System|Setting|Multimedia|Metadata|Extract|XMP|DetRights|http://purl.org/dc/elements/1.1/:rights;http://ns.adobe.com/xap/1.0/rights/:UsageTerms
```

Notice that the last entry looks up both the Dublin Core rights attribute and the XMP Rights Management usage attribute.

When using the Update Resource tool (*Multimedia->Update Resource*) for a selection of records, the metadata extraction Registry entries are not used. If, however, you update the current record, the extraction does take place.

Limiting Colours computation

When using the Update Resource tool (*Multimedia->Update Resource*) all image characteristics for the master image, thumbnail and derivatives are updated. Any metadata embedded in the master image is also updated in the *EXIF*, *IPTC* and *XMP* tabs.

When upgrading to EMu 3.2.03 the Update Resource tool must be run for all records in the Multimedia repository that contain images to populate the new metadata and properties fields. When updating the image attributes the most expensive value in terms of computing time is the *Colours* value. This value displays the number of unique colours in the image. To calculate the number each pixel in the image must be examined, so for very large images, that is those with a large number of pixels, the time taken can be considerable (up to 2-3 minutes for an 8,000 by 8,000 pixel image).

If a large number of large images need to be updated, considerable time may be taken. In order to reduce this time a Registry entry has been added that allows a threshold to be set over which the number of colours is not calculated (the value is left empty). The format of the entry is:

```
User|user|Setting|Multimedia|Colours|Maximum Size|value
Group|group|Setting|Multimedia|Colours|Maximum Size|value
Group|Default|Setting|Multimedia|Colours|Maximum Size|value
System|Setting|Multimedia|Colours|Maximum Size|value
```

where *value* is the threshold in kilobytes (KB) over which the colours value is no longer calculated. For example, the entry:

```
System|Setting|Multimedia|Colours|Maximum Size|1024
```

will result in all files over 1 megabyte (1024 Kb) not having their number of unique colours value calculated. Note that if the *Multimedia->Update Resource->Current Record* command is executed the number of colours is always calculated regardless of this Registry setting. This allows the value to be computed if required for an image that is over the threshold.