MEDIARICH™

Equilibrium® MediaRich™ Metadata Specification

Version 1.0

Introduction

If your firm uses an image management system (IMS) or content management system (CMS) for storing and retrieving digital assets, you need to preserve and update the metadata within generated image files. MediaRich provides support for the most popular metadata formats: IPTC, Exif, and XMP.

Supported formats

MediaRich fully supports loading, saving and merging IPTC, Exif, and XMP metadata for JPEG, TIFF, and Photoshop™ files.

The MediaRich for SharePoint™ product currently supports only IPTC and Exif metadata for JPEG, TIFF, and Photoshop files.

IPTC

IPTC Metadata, also known as IPTC Comments or File Info, was developed for press photographers who needed to attach comments to images when submitting them electronically. The amount of information that can be added to each image is extensive, but usually consists of general information such as a caption, the place and date a photo was taken, comments and copyright. Many digital cameras support IPTC data.

A detailed schema is provided for the IPTC documents constructed by MediaRich.

Exif

EXIF is a metadata standard for image files, used widely by digital cameras. The EXIF 2.2 specification can be found at http://www.exif.org/specifications.html.

A detailed schema is provided for the Exif documents constructed by MediaRich.

XMP

MediaRich also supports loading XMP metadata from the following file formats: Illustrator™, EPS, GIF, PDF, and PNG. This metadata is available to the script as a metadata XML document. The XMP specification can be found at http://partners.adobe.com/public/developer/xmp/sdk/index.html.

The XMP metadata document conforms to the schema defined by Adobe®.

Low-level Metadata Interface

Two MediaScript objects provide support for metadata: the `Media` object and the `_MR_Metadata` object.

- The `Media` object provides support for loading and saving metadata along with the image contents.
- The `_MR_Metadata` object provides support for loading and merging just the metadata contained within the files, without loading the image data. This allows the scripter to modify the metadata within compressed files without decompressing and recompressing the image data.

The Media object

The `load` command of the `Media` object loads and attaches Exif, IPTC, and XMP metadata if the `LoadMetadata` parameter to the load command is specified as `true`. Consider the following example:

```
var image = new Media();
image.load(name @ "myimage.jpg", loadMetadata @ true);
```

This constructs XML documents for any Exif, IPTC, or XMP metadata contained in the file and attaches them to the `Media` object. This metadata is accessible by the scripter using the `getMetadata` command of the `Media` object:

```
var metaDoc = image.getMetadata("IPTC");
```

This metadata document can be processed and edited. To modify the metadata attached to the image, use the `setMetadata` command, as in the following example:

```
image.setMetadata("Exif", myExifData);
```

The metadata names used with the `getMetadata` and `setMetadata` commands are "Exif", "IPTC", and "XMP" for the Exif, IPTC, and XMP metadata documents, respectively. These names are *case-sensitive*.

Finally, the `save` command automatically saves any metadata attached to the document unless the `SaveMetadata` parameter is specified as "false" or the file type specified does not support metadata. The following example saves the attached metadata:

```
image.save(type @ "jpeg");
```

The _MR_Metadata object

The `_MR_Metadata` object supports extracting metadata from supported file formats without loading the image data. It also supports merging new metadata into existing files without the need to interpret or decompress the image data. The `_MR_Metadata` object has two methods: `save` and `load`. In addition, the `_MR_Metadata` object can be used as the MediaScript response object allowing the script to stream back a file with modified metadata.

The `_MR_Metadata` constructor takes a file name and an optional file type, as illustrated in the following example:

```
var metaObj = new _MR_Metadata("myimage.jpg", "jpeg");
```

If the file has a valid extension, the file type can be omitted.

The `_MR_Metadata` object `save` command provides a single object as a parameter, similar to the `Media` object `save` command. You can specify this parameter as an object, or by using the "@" notation.

The parameters for the save command object are `exif`, `iptc`, `xmp`, and `name`. The file type of the saved file is always the same as the file type of the original image.

```
metaObj.save( exif @ myExifDoc, iptc @ null, xmp @ null, name @
"newFile"); or

var saveObj = new Object();
saveObj.iptc = null;
saveObj.xmp = null;
saveObj.exif = myExifDoc;
saveObj.name = "newFile";
metaObj.save(saveObj);
```

If any of the `exif`, `iptc`, or `xmp` parameters are omitted, existing metadata of that type in the file are transferred to the output file. If any of the `exif`, `iptc`, or `xmp` parameters are specified as `null`, existing metadata of that type is omitted in the output. Otherwise, IPTC and XMP data is replaced using the specified data, and writable Exif tags are replaced. *NOTE:* The Exif camera data tags are never replaced.

High-level Metadata Interface for Exif and IPTC

Two MediaScript objects are provided to simplify the tasks of getting and setting individual metadata items. These objects provide high-level support for IPTC and Exif metadata. Each of these objects has a similar format, providing set<Tag> methods and get<Tag> methods which set and get individual metadata fields, respectively. <Tag> represents the name of the metadata tag to set or get.

The following section describes the general structure and common methods for both the IPTCMetadata object and the ExifMetadata object. Later sections provide descriptions of the set<Tag> and get<Tag> methods for the `IPTCMetadata` and `ExifMetadata` objects.

Common metadata methods

The `IPTCMetadata` and `ExifMetadata` objects have several common methods allowing the scripter to create documents, specify metadata for existing documents, extract a string representation of the XML document, and validate the document. The following table provides a list of these operations with a brief description:

`IPTCMetadata(validate)`	Constructs a blank IPTC metadata document. If validate is true, the document is automatically validated in set<tag> methods.
`ExifMetadata(validate)`	Constructs a blank Exif metadata document.
`loadFromFile(filename)`	Loads metadata object with data from the specified file.
`loadFromMedia(media)`	Loads metadata object with data from the specified media.
`loadFromXML(xmlString)`	Loads metadata object with data from the specified XML string.
`blankDocument()`	Loads metadata object with a valid blank document.
`validate()`	Validates the document to the appropriate schema.
`isEmpty()`	Returns true if the document is empty.

Note: Each metadata constructor constructs a *blank* metadata document of the appropriate type. This document is not empty, and is a valid XML document for the appropriate metadata schema. However, `loadFromFile` and `loadFromMedia` leave the document in an empty state if the file contains no metadata of the appropriate type.

```
var metadata = new IPTCMetadata();
var empty = metadata.isEmpty(); // returns false
metadata.loadFromFile("img.jpg");
if (!metadata.isEmpty())
{

     // do something with metadata
}
else
{

     // file did not contain metadata.
}
```

Alternatively, you can simply use the get<Tag> methods, which will return null if the document is empty.

IPTCMetadata object

The following table provides a list of the methods that can be used to get and set metadata values for IPTC metadata and a brief description of each method. Refer to the schema (IPTC.xsd) in the `//Shared/Originals/Sys` folder for the required format for each of the IPTC fields. For a complete description of each IPTC metadata field, please consult the IPTC news-photo metadata specification available at http://www.iptc.org under the title "Digital Newsphoto Parameter Record".

Note: The notation "(string …)" indicates that multiple values can be specified as arguments to the method.

`getVersion()`	Returns the version field.
`setVersion(string)`	Sets the version field.
`getObjectTypeReference()`	Returns the object type reference field.
`setObjectTypeReference(string)`	Sets the object type reference field.
`getObjectAttributeReference()`	Returns an array of attribute references.
`setObjectAttributeReference(string, …)`	Sets attribute references.
`addObjectAttributeReference(string, …)`	Adds an attribute references to the list.
`setObjectAttributeReferenceArray(array)`	Sets a group of attribute references from an Array.
`getObjectName()`	Returns the object name.
`setObjectName(string)`	Sets the object name.
`getEditStatus()`	Returns the edit status.
`setEditStatus(string)`	Sets the edit status.
`getEditorialUpdate()`	Returns the editorial update code.
`setEditorialUpdate(string)`	Sets the editorial update code.
`getUrgency()`	Returns the urgency code.
`setUrgency(string)`	Sets the urgency code.
`getSubjectReference()`	Returns an array of subject references.
`setSubjectReference(string, …)`	Sets subject references.
`addSubjectReference(string, …)`	Adds subject references to the list.
`setSubjectReferenceArray(array)`	Sets a group of subject references from an Array.
`getCategory()`	Returns the category code.
`setCategory(string)`	Sets the category code.
`getSupplementalCategory()`	Returns the supplemental category array.

`setSupplementalCategory(string, …)`	Sets supplemental categories.
`addSupplementalCategory(string, …)`	Adds a value to the supplemental category list.
`setSupplementalCategoryArray(array)`	Sets a group of supplemental categories as an Array.
`getFixtureIdentifier()`	Returns the fixture identifier code.
`setFixtureIdentifier(string)`	Sets the fixture identifier code.
`getKeywords()`	Returns an array of keywords.
`setKeywords(string, …)`	Sets keywords.
`addKeywords(string, …)`	Adds keywords to the list
`setKeywordsArray(array)`	Sets keywords from an Array.
`getContentLocation()`	Returns an array of content location objects. Each object has two properties: ContentLocationName and ContentLocationCode.
`getContentLocationName(which)`	Returns the ContentLocationName subfield of the ContentLocation tag indexed by which.
`getContentLocationCode(which)`	Returns the ContentLocationCode subfield of the ContentLocation tag indexed by which.
`setContentLocation(name, code)`	Sets the ContentLocation tag to the specified name and code.
`addContentLocation(name, code)`	Adds the ContentLocation specified by name and code to the ContentLocation list.
`getReleaseDate()`	Returns the ReleaseDate and ReleaseTime tags as a MediaScript Date object.
`setReleaseDate(date)`	Sets the ReleaseDate and ReleaseTime tags from a MediaScript Date object.
`setReleaseTime(string)`	Sets only the ReleaseTime field.
`getExpirationDate()`	Returns the ExpirationDate and ExpirationTime fields as a MediaScript Date object.
`setExpirationDate(date)`	Sets the ExpirationDate and ExpirationTime fields from a MediaScript Date object.
`setExpirationTime(string)`	Sets only the ExpirationTime field.
`getSpecialInstructions()`	Returns the SpecialInstructions field.
`setSpecialInstructions(string)`	Sets the SpecialInstructions field.
`getActionAdvised()`	Returns the ActionAdvised field.
`setActionAdvised(string)`	Sets the ActionAdvised field.
`getReference()`	Returns an array of Reference object for the Reference field. Each reference object has a ReferenceService, ReferenceDate, and ReferenceNumber property.

`getReferenceService(which)`	Returns the ReferenceService property of the Reference element indexed by which.
`getReferenceDate(which)`	Returns the ReferenceDate property of the Reference element indexed by which as a MediaScript Date object.
`getReferenceNumber`	Returns the ReferenceNumber property of the Reference element indexed by which.
`setReference(service, date, number)`	Sets the Reference element to the reference specified by service, date and number. Date must be a MediaScript Date object.
`addReference(service, date, number)`	Adds a Reference element to the list using the specified service, date and number. Date must be a MediaScript Date object.
`getDateCreated()`	Returns the DateCreated and TimeCreated fields as a MediaScript Date object.
`setDateCreated(date)`	Sets the DateCreated and TimeCreated fields from a MediaScript Date object.
`setTimeCreated(string)`	Sets the TimeCreated field.
`getDigitalCreationDate()`	Returns the DigitalCreationDate and DigitalCreationTime fields as a MediaScript Date object.
`setDigitalCreationDate(date)`	Sets the DigitalCreationDate and DigitalCreationTime fields from a MediaScript Date object.
`setDigitalCreationTime(string)`	Sets the DigitalCreationTime field.
`getOriginatingProgram()`	Returns the OriginatingProgram field.
`setOriginatingProgram(string)`	Sets the OriginatingProgram field.
`getProgramVersion()`	Returns the ProgramVersion field.
`setProgramVersion(string)`	Sets the ProgramVersion field.
`getObjectCycle()`	Returns the ObjectCycle field.
`setObjectCycle(string)`	Set the ObjectCycle field.
`getByLine()`	Returns an array of ByLine objects, each of which contains a ByLineWriter and a ByLineTitle property.
`getByLineWriter(which)`	Returns the ByLineWriter property of the ByLine element specified by which.
`getByLineTitle(which)`	Returns the ByLineTitle property of the ByLine element specified by which.
`setByLine(writer, title)`	Sets the ByLine element to the specified writer and title.
`addByLine(write, title)`	Adds an element to ByLine for the given writer and title.

`getCity()`	Returns the City element.
`setCity(string)`	Sets the City element.
`getSublocation()`	Returns the Sublocation.
`setSublocation(string)`	Sets the Sublocation.
`getState()`	Returns the State (Province).
`setState(string)`	Sets the State (Province).
`getCountryCode()`	Returns the CountryCode.
`setCountryCode(string)`	Sets the CountryCode.
`getCountryName()`	Returns the CountryName.
`setCountryName(string)`	Sets the CountryName.
`getOriginalTransmissionReference()`	Returns the OriginalTransmissionReference.
`setOriginalTransmissionReference(string)`	Sets the OriginalTransmissionReference.
`getHeadline()`	Returns the Headline.
`setHeadline(string)`	Sets the Headline.
`getCredit()`	Returns the Credit.
`setCredit(string)`	Sets the Credit field.
`getSource()`	Returns the Source field.
`setSource(string)`	Sets the Source field.
`getCopyrightNotice()`	Returns the Copyright field.
`setCopyrightNotice(string)`	Sets the Copyright field.
`getContact()`	Returns an array of Contact elements.
`setContact(string, …)`	Sets Contact elements.
`addContact(string, …)`	Adds Contact elements.
`setContactArray(array)`	Sets Contact element from an Array.
`getCaption()`	Returns the Caption element.
`setCaption(string)`	Sets the Caption element.
`getWriter()`	Returns an array of writer elements.
`setWriter(string, …)`	Sets Writer elements.
`addWriter(string, …)`	Adds Writer elements.
`setWriterArray(array)`	Sets Writer elements from an array.
`getImageType()`	Returns the ImageType.
`setImageType(string)`	Sets the ImageType.

`getImageOrientation()`	Returns the ImageOrientation.
`setImageOrientation(string)`	Sets the ImageOrientation.
`getLanguageIdentifier()`	Returns the LanguageIdentifier.
`setLanguageIdentifier(string)`	Sets the LanguageIdentifier.

Exif Metadata object

The following methods may be used to get and set metadata values for Exif metadata. Refer to the schema (Exif.xsd) in the `//Shared/Originals/Sys` folder for the required format for each of the Exif fields, as only a brief description is provided here. For a complete description of each Exif metadata field, consult the Exif metadata specification available at http://www.exif.org.

Where possible, these values are converted into string-valued representations as defined in the Exif specification.

IFD0	
`getImageDescription()`	Returns the image description.
`setImageDescription(string)`	Sets the image description.
`getOrientation()`	Returns the image orientation.
`setOrientation(string)`	Sets the image orientation.
`getSoftware()`	Returns the software description.
`setSoftware(string)`	Sets the software description.
`getArtist()`	Returns the artist.
`setArtist(string)`	Sets the artist.
`getDateTime()`	Returns the DateTime field as a MediaScript Date object.
`setDateTime(date)`	Sets the DateTime field from a MediaScript Date object.
`getPhotographerCopyright()`	Returns the photographer copyright.
`setPhotographerCopyright(string)`	Sets the photographer copyright.
`getEditorCopyright()`	Returns the editor copyright.
`setEditorCopyright(string)`	Sets the editor copyright.
`getMake()`	Returns the camera make.
`getModel()`	Returns the camera model.
`getImageWidth()`	Returns the image width.
`getImageLength()`	Returns the image height.
`getBitsPerSample()`	Returns the number of bits per sample.
`getCompression()`	Returns the compression type.
`getPhotometricInterpretation()`	Returns the photometric interpretation.

`getPlanarConfiguration()`	Returns the planar configuration.
`getYCbCrSubSampling()`	Returns the YCbCr sub-sampling.
`getYCbCrPositioning()`	Returns the YCbCr positioning.
`getXResolution()`	Returns the horizontal resolution.
`getYResolution()`	Returns the vertical resolution.
`getResolutionUnit()`	Returns the resolution unit.
`getWhitePoint()`	Returns the white point.
`getPrimaryChromaticities()`	Returns the primary chromaticities.
`getYCbCrCoefficients()`	Returns the YCbCr coefficients.
`getReferenceBlackWhite()`	Returns the ReferenceBlackWhite value.
IFDExif	
`getVersion()`	Returns the Exif version.
`setVersion(string)`	Sets the Exif version (default = 2.1)
`getFlashPixVersion()`	Returns the flashpix version.
`setFlashPixVersion(string)`	Sets the flashpix version
`getUserComment()`	Returns the user comment.
`setUserComment(string)`	Sets the user comment.
`getColorspace()`	Returns the colorspace.
`getPixelXDimension()`	Returns the width.
`getPixelYDimension()`	Returns the height.
`getComponentsConfiguration()`	Returns the ComponentsConfiguration value.
`getCompressedBitsPerPixel()`	Returns the approx. number of compressed bits per pixel.
`getRelatedSoundFile()`	Returns the name of a related sound file.
`getDateTimeOriginal()`	Returns a MediaScript Date object for the original date/time.
`getDateTimeDigitized()`	Returns a MediaScript Date object for the digitized date/time.
`getSubSecTime()`	Returns the sub-second time offset.
`getSubSecTimeOriginal()`	Returns the sub-second time offset for the original.
`getSubSecTimeDigitized()`	Returns the digitized sub-second time offset.
`getExposureTime()`	Returns the exposure time.
`getShutterSpeedValue()`	Returns the shutter speed in seconds.
`getApertureValue()`	Returns the aperture value as an F-number.
`getBrightnessValue()`	Returns the brightness value.
`getExposureBiasValue()`	Returns the exposure bias value.
`getMaxApertureValue()`	Returns the maximum aperture value as an F-number.

`getSubjectDistance()`	Returns the subject distance in meters.
`getMeteringMode()`	Returns the metering mode.
`getLightSource()`	Returns the light source.
`getFlash()`	Returns true if flash was used.
`getFocalLength()`	Returns the focal length.
`getFNumber()`	Returns the F-number.
`getExposureProgram()`	Returns the exposure program.
`getSpectralSensitivity()`	Returns the spectral sensitivity.
`getISOSpeedRatings()`	Returns the ISO film speed.
`getOECF()`	Returns the OECF value.
`getFlashEnergy()`	Returns the flash energy.
`getSpatialFrequencyResponse()`	Returns the spatial frequency response.
`getFocalPlaneXResolution()`	Returns the focal-plane horizontal resolution.
`getFocalPlaneYResolution()`	Returns the focal-plane vertical resolution.
`getFocalPlaneResolutionUnit()`	Returns the focal-plane resolution unit.
`getSubjectLocation()`	Returns the subject location.
`getExposureIndex()`	Returns the exposure index.
`getSensingMethod()`	Returns the sensing method.
`getFileSource()`	Returns the file source.
`getSceneType()`	Returns the scene type.
`getCFAPattern()`	Returns the CFA pattern.