

DIAGRAM Center Working Paper – April 2011

Potential Use of Image Description Metadata for Accessibility

This working paper outlines the current barriers and opportunities for providing image description within publishing tools, technologies, and production processes. Ongoing discussion with standards groups, Adobe and other tool developers is exploring the feasibility of extending support for alt text and longdesc within image formats and authoring tools.

Background.....	1
Image Metadata Specifications and Related Standards Organizations.....	3
Table 1 – Typical Camera Metadata.....	3
Table 2 – Metadata Formats Supported in Image Formats.....	7
Opportunities for Image Description in Metadata Specifications.....	7
Table 3 – Metadata Working Group Description Field Cross References.....	8
Metadata Handling within Existing Tools	10
Image Creation and Editing Tools.....	11
Image Management Tools.....	17
Content Publishing Tools.....	19
Table 4 — Product Features Relative to Embedded Image Metadata.....	22
Summary.....	23
Useful References.....	23

Background

The use of images in reading material presents significant challenges to blind and visually impaired (B/VI) readers. Electronic formats for these reading materials, including on-line websites, electronic books (e-books), etc., have allowed B/VI readers much greater access to text materials. Some software applications can enlarge screen images for VI users, while screen reader applications, running between the operating system and the application layers, convert underlying text data and controls to the spoken word or to braille. A number of screen reader technologies are in wide use today, and they are compatible with a wide range of applications, including word processors, spreadsheets, web browsers, e-book readers, and others.

Reading materials do not, however, contain only textual information. Many contain pictures, graphs, tables, etc. which are contained in image formats with no text equivalents. Rendering of such implicitly non-textual information for B/VI readers requires additional information. Some distribution formats (such as HTML, rendered by web browsers and DAISY, rendered by digital book

readers) contain specifications that allow an image to have text data associated with it. For example, the “alt” attribute (sometimes referred to as “@alt”) of an <image> element in a DAISY e-book can be used to contain a short description of the underlying image. The current DAISY specification also has a <prodnote> element, which can serve the same function. As an example, consider the following DAISY snippet for an image:

```

```

As the B/VI user scans the document, this markup would be intercepted by a screen reader and voiced (for example) as “image, student holds beaker of liquid over a Bunsen burner”. Without the “alt” attribute, the screen reader might simply voice “image, picture.gif”, which does not convey useful information to a B/VI user.

To provide access to images, the author or some other contributor has to look at the picture and create the “alt” text for that specific image element in the document. Sometimes the publication software has built-in functionality for adding the “alt” tag in the source, or project, format. At other times, the publishing software does *not* have such functionality, and the “alt” tag must be manually added into the end product, if possible. In the case of DAISY books, the internal format is human readable, and an “alt” tag can be added with a simple text editor (although this is a challenging workflow). Multiply this process of creating image descriptions by the hundreds or even thousands of images in a book, and multiply again by the number of target book formats, which might include PDF, ePub, DAISY, HTML, etc. The result is a very large matrix of potential incompatibilities or problematic workflows: similar problems arise for almost all target platforms and workflows.

Compounding the problem of multiple input and output formats in the workflow is the challenge of authoring meaningful description; the people creating the technical layout for the books are usually quite facile with the technical publishing tools but are not typically the subject matter experts who would be most capable of authoring useful descriptions. And it is often not practical for those subject matter experts to become expert publishing-tool users.

In order to address some of the workflow problems discussed above, and to expand the amount of image description in electronic publications, one possibility is to embed the image description text directly inside the image. Tools might be able to put the equivalent of “alt” text directly inside an image, stored as metadata. When the image is imported into the publishing tools, or extracted from the CMS or other data store, the embedded description metadata could be extracted by the publishing mechanism and rewrapped in the target data format, whether ePub, DAISY, PDF, HTML, etc.

The purpose of this paper is to survey the currently available mechanisms, data formats, and tools which could be leveraged for such use. Topics outlined:

- Image Metadata Specifications and Related Standards Organizations
- Opportunities for Image Description in Metadata Specifications
- Metadata Handling within Existing Tools

Image Metadata Specifications and Related Standards Organizations

There are several image metadata systems in wide use today. At one end of the workflow is image capture; even the most basic digital cameras typically record and store some metadata inside each image. That data is related to camera settings, such as the type of lens, the aperture, shutter speed, film ISO, etc. Some higher-end cameras also record spatial/location information using built-in GPS technology. Such camera- and hardware-related metadata come in several formats, including EXIF, IPTC, etc. At the next stage of the workflow, image manipulation and publication, there are technologies, such as Adobe's XMP, which attempt to create metadata supersets to extend the hardware metadata to include other production metadata such as content tracking, rights management, etc. Table 1 below shows some typical camera metadata tags and sample values. It is not an exhaustive list, but is informative.

Table 1 – Typical Camera Metadata

Tag	Sample Value
Exif Version	Exif Version 2.1
Manufacturer	CASIO
Model	QV-4000
Software	Ver1.01
Compression	JPEG compression
Date and Time	2003:08:11 16:45:32
Exposure Time	1/659 sec.
FNumber	f/4.0
Focal Length	20.1 mm

The following is a short description of some of the major image metadata formats and their respective standards bodies and supporting organizations, which maintain efforts to formalize and consolidate the standards. These include formal image specifications (such as EXIF), pure metadata specifications (such as Dublin Core and XMP), and function specifications (such as Metadata Working Group).

EXIF

The Exchangeable Image File Format (EXIF) is the standard for image file storage for digital still cameras. It is maintained by the Japan Electronics and Information Technology Industries Association (JEITA) as a standard way of storing images created by digital cameras as well as metadata about the images. Most digital cameras now use the EXIF format. EXIF image metadata can be stored in TIFF (for uncompressed) and JPEG (for compressed) formatted images. The format is part of the DCF (Design rule for Camera File system) standard created by [JEITA](http://www.jeita.or.jp/english/) to encourage interoperability between imaging devices.

For more information, see:

<http://exif.org>

<http://www.jeita.or.jp/english/>

IPTC

The International Press Telecommunications Council (IPTC), based in London, UK, develops and maintains technical standards for news exchange that are used by virtually every major news organization in the world. The IPTC Information Interchange Model (IPTC-IIM) Version 4 metadata standard captures information that is important to the activities of news gathering, reporting, and publishing and is supported by the International Press Telecommunications Council and the Newspaper Association of America. These information records are commonly referred to as IPTC tags. The use of embedded IPTC tags in image file formats became widespread with the use of the Adobe Photoshop tool for image editing. IPTC metadata can be stored in TIFF and JPEG format images.

For more information, see:

<http://www.iptc.org>

http://www.iptc.org/std/photometadata/specification/IPTC-PhotoMetadata-201007_1.pdf

Dublin Core

The mission of the Dublin Core Metadata Initiative (DCMI) is to provide standards to facilitate the finding, sharing and management of information. DCMI develops and maintains international standards for describing resources, supports a worldwide community of users and developers, and promotes widespread use of Dublin Core solutions.

Dublin Core was developed specifically to address the difficulty in finding resources on the web. It defines a core set of fields designed to describe resources. It has been extended into many domains, and is in wide use in the media industry. While not specifically an internal image metadata format, Dublin Core metadata is specifically supported in the XMP specification. XMP-formatted metadata can include the Dublin Core namespace, and thus Dublin Core-tagged metadata can be included inside image files. (See more on XMP in next section.)

For more information, see:

<http://www.dublincore.org>

Adobe XMP

The Extensible Metadata Platform (XMP) is a standard metadata format, developed by Adobe, for the creation, processing, and interchange of metadata in a variety of applications. XMP also defines how the data model is serialized (converted to a byte stream), and embedded within an image file.

In Adobe's XMP Specification, Part 3, the location of XMP metadata is defined across a large array of image file formats, including TIFF, JPEG, JPEG 2000, PNG, GIF, and PDF. Additionally, for file formats that have no support for embedded XMP data, or that are not image file themselves (such as HTML, etc.,) this data can be stored in external .xmp sidecar files.

For more information about XMP, see the Adobe Web site at
<http://www.adobe.com/>
<http://www.adobe.com/products/xmp>
[http://www.images.adobe.com/www.adobe.com/content/dam/Adobe/en/devnet/xmp/pdfs/
XMPSpecificationPart3.pdf](http://www.images.adobe.com/www.adobe.com/content/dam/Adobe/en/devnet/xmp/pdfs/XMPSpecificationPart3.pdf)

Metadata Working Group

The Metadata Working Group (MWG) is a consortium of companies in the digital media industry, focused on the following goals: preservation and seamless interoperability of digital image metadata interoperability and availability to all applications, devices, and services. The MWG publishes technical specifications that describe how to effectively store metadata in digital media files. These royalty-free specifications are made available to software developers, manufacturers and service providers so that they may create products that use metadata in a consistent way, and that allow consumers to better describe, organize and find their media. Where possible, these specifications rely on existing standards, and aim to create a unified and cohesive approach to applying these standards.

Of particular interest in the MWG guidelines is the section on data duplication and preservation. (Note that in the description below, use of the term “Consumer” is the MWG’s nomenclature for software that only reads metadata from an image file, but doesn’t otherwise transform that data or even export it to other output products.) Specifically, the guidelines requires two things (page 13 of [mwg_guidance.pdf](#)):

2.1.2 Changer A **Changer** application first reads metadata from the image file and then writes new or modified metadata back to the same file. The rules for an application in **Changer** role are:

- Deletion of metadata **MUST** only be done with specific intent.
- It **SHOULD** obey rules for **Consumer** applications when reading metadata.
- It **SHOULD** keep all forms of metadata it modifies in sync with each other.

The first rule is about preservation of metadata, which is a high priority. Descriptive metadata, information added by a user, **MUST** only be deleted by explicit user intent. Special attention **SHOULD** be paid to formally sensitive information such as a copyright. Non-descriptive metadata **MAY** be deleted by explicit user intent, **SHOULD** be modified or deleted if it is known to be inaccurate, otherwise it **SHOULD** be preserved. The second rule comes from the fact that, almost always, the **Changer** application is also a **Consumer** application so it must also observe all **Consumer** application rules. The third rule states, that whenever, the **Changer** application writes new metadata fields to the file, it must keep different forms existing in the file, e.g. Exif, IPTC-IIM and XMP, in sync. The first rule and third rule also apply to a changer that wishes to not write all forms of a metadata item. Writing one form and deleting other forms is a legitimate intentional deletion, done to not have unsynchronized forms in the file.

2.1.3 Consumer A **Consumer** application only reads metadata from the image file. It may use metadata for display purposes, searching, content organization, etc. but it never modifies the metadata in the file itself. General rules for **Consumer** application are:

- It **MUST** reconcile between different forms of metadata in the image.
- It **MUST** use metadata according to the semantics defined for each field.

The first rule says that a **Consumer** application must process metadata according to policies defined in this document and then only the reconciled data is used further before it is presented to the end- user. This may involve, for example resolving possibly conflicting information between Exif, IPTC-IIM and XMP versions of a metadata field existing in the image file. The second rule says that applications must understand the semantics of desired metadata fields and use them appropriately. For example, in order to reconcile different forms of metadata, the application must know the semantics of the metadata in question. In summary, the **Consumer** application treats image files as read-only so the state of the metadata remains unchanged. Tools designed to display technical details about the format and content of the file's metadata, but not intended to primarily express metadata semantic meaning, are not required to be compliant **Consumer** applications.

This is quite important, as it implies that well-behaved applications should allow image metadata to travel safely up and down the production chain; this could be beneficial if image descriptions are included in metadata. However, it does *not* require that image metadata be included in all output formats for the images. This is discussed in more detail in the section *"Metadata Handling within Existing Tools."*

For more information, see:

<http://www.metadataworkinggroup.org>

<http://www.metadataworkinggroup.org/specs>

http://www.metadataworkinggroup.org/pdf/mwg_guidance.pdf)

SVG

Scalable Vector Graphics (SVG) is a widely deployed, royalty-free, vector-based graphics format which uses XML as the underlying data storage format in the file. SVG is developed and maintained by the W3C SVG Working Group. The SVG format can be thought of as a set of drawing commands which are stored in the image file. When taken as a set, these commands constitute a renderable image. An application which supports SVG images then executes those commands to render the image. The following is a sample snippet of SVG XML which draws a rectangle:

```
<svg width="100%" height="100%" version="1.1" xmlns="http://www.w3.org/2000/svg">
  <g>
    <rect width="300" height="100" style="fill:rgb(0,0,255); stroke-width:1; stroke:rgb(0,0,0)"/>
  </g>
</svg>
```

In addition to describing vector-based graphics, SVG also includes specifications for storing rasterized images by describing RGB values for individual pixels. While not as efficient as (for example) TIFF or JPEG for storage of such images, the readability of such a file can be quite useful for search and other purposes like geomatics, etc. Because SVG is an XML format, its raw data can be viewed (and even authored) in a simple text editor. This distinguishes it from the raster-based image formats such as TIFF, JPEG, etc. which are the domain of EXIF and XMP metadata.

While SVG’s XML elements are primarily defined for drawing primitives, there are a couple of elements which can be thought of as metadata. Use of those elements for image description is discussed further in the next section.

For more information, see:
<http://www.w3.org/Graphics/SVG/>

As this summary shows, several image metadata format specifications are currently in use in the industry, but they are quite disparate, having been developed at different times and for different purposes. And while the standards bodies and organizations are keeping their respective specifications active and dynamic, the Metadata Working Group is, fortunately, taking an active role in nudging the industry toward, if not a single specification, at least a rational method of combining and tracking them. As will be shown in subsequent sections, EXIF, IPTC, and XMP are fairly well unified for some image-editing applications. SVG does not fall under that same unifying umbrella since it takes a wholly different approach to graphic formatting. It should be noted that not all metadata formats are supported in all image file formats. Table 2 below summarizes the various metadata formats described above versus the image file formats into which they can be embedded.

Table 2 – Metadata Formats Supported in Image Formats

Metadata format	Supported in image formats
EXIF	TIFF, JPEG
IPTC	TIFF JPEG
XMP	TIFF, JPEG, JPEG 2000, PNG, GIF, PDF
SVG	SVG

Opportunities for Image Description in Metadata Specifications

The previous section discussed existing metadata standards currently in use by the industries using images. This section highlights specific fields within those metadata specifications which might be harnessed to carry image descriptions for accessibility.

Table 3, below, from the Metadata Working Group’s *Guidelines for Handling Metadata Section 5.2 Description*, identifies the cross-referencing of description fields across several metadata formats. These fields could be used as potential image description fields in three of the metadata specifications discussed in the previous section.

Table 3 – Metadata Working Group Description Field Cross References

Description	A textual description of the contents shown in the image
Representation	Information for the description property is available in the following properties: Exif ImageDescription (270, 0x010E) IPTC Caption (IIM 2:220, 0x0278) XMP (dc:description[x-default])
Type	See respective specifications
Value	See respective specifications
Guidance	See chapter “Handling Exif/TIFF, IPTC-IIM and XMP metadata”
Restrictions	Length limitation in IPTC-IIM is 2000 bytes
Notes	Exif ImageDescription, IPTC Caption, and XMP (dc:description) are mapped together. In XMP, the description can be represented in multiple languages. In this document only the “x-default” value will be discussed and used. Clients MAY support the full range of localized values.

It should be noted that while the Metadata Working Group’s guideline above indicates that all of the “description” fields in the various metadata specifications can be easily unified, each specification is independent of the others.

What follows is a compilation of the definitions of the field identified above, but extracted from each specification.

EXIF

ImageDescription

A character string giving the title of the image. It may be a comment such as "1988 company picnic" or the like. Two-byte character codes cannot be used. When a 2-byte code is necessary, the Exif Private tag *UserComment* is to be used.

Tag = 270 (10E.H)

Type = ASCII

Count = Any

Default = None

For more information see:

<http://exif.org/Exif2-1.PDF>

IPTC

The IPTC defers to the Metadata Working Group, (see page 28 of above document):
For the exact processing while synchronising metadata values please read the Guidelines for Handling Image Metadata from the Metadata Working Group. The document can be obtained from <http://www.metadataworkinggroup.org/specs/>

For more information see:

http://www.iptc.org/std/photometadata/specification/IPTC-PhotoMetadata-201007_1.pdf

Dublin Core

Term Name: *description*

URI: <http://purl.org/dc/elements/1.1/description>

Label: Description

Definition: An account of the resource.

Comment: Description may include but is not limited to: an abstract, a table of contents, a graphical representation, or a free-text account of the resource.

Also note that the DAISY consortium already makes a normative reference to the use of Dublin Core Metadata (section 3.2 of the Specification for Digital Talking Book):

- dc:Description
 - Content: Plain text describing the publication's content.
 - Occurrence: Optional

For more information see:

<http://dublincore.org/documents/dces/>

<http://www.daisy.org/z3986/2005/Z3986-2005.html>

Scalable Vector Graphics (SVG)

As mentioned in the previous section, SVG is a full specification for drawing images using vector graphics commands in an XML format. SVG has some built-in support for description metadata. Specifically, it defines both a <title> element and a <desc> element that can be used to contain descriptive metadata. Both of these elements can only contain plain text so they cannot be used for elaborate description metadata. The <title> element in SVG serves roughly the same function as it does in the HTML format, and there should only be one of them. Screen readers, such as JAWS, currently recognize the <title> element in an SVG file and voice it.

The <desc> element is potentially more flexible for description, since there can be multiple instances of this element. For example, in a complex bar graph, there could be a <desc> element for each bar of the graph, thereby potentially allowing a screen reader to read meaningful information while permitting the user to navigate through it in chunks. The <desc> tag of an SVG is currently only read by a single device, the IVEO Viewer (which also reads the <title> element). It is not currently read by general-use screen readers such as JAWS.

Modifying the original SVG example from the previous section:

```
<svg width="100%" height="100%" version="1.1" xmlns="http://www.w3.org/2000/svg">
  <g>
    <rect width="300" height="100" style="fill:rgb(255,0,0); stroke-width:3;
    stroke:rgb(0,0,255)"/>
    <title>rectangle</title>
    <desc>A red rectangle with a blue border</desc>
  </g>
</svg>
```

For more information see:

<http://www.w3.org/TR/SVG-access/>

<http://www.w3.org/TR/SVG/struct.html#DescriptionAndTitleElements>

<http://manuals.viewplus.com/en/view/7/55/>

Thus, although there are several widely disparate image metadata formats, specifications, and efforts, there is enough commonality amongst them and enough overlap of some key fields that it should be possible to develop a coherent workflow to harness certain metadata fields for use as image description containers for accessibility. The next section looks at how existing tools treat image metadata.

Metadata Handling within Existing Tools

For image creation and editing, image management and content publishing tools, we are interested in four questions regarding metadata handling:

1. Does it allow direct creation or authoring of embedded image metadata?
2. Does it preserve all upstream embedded image metadata in its export products?
3. Does it automatically populate downstream accessibility features in export products?
4. Does it allow for manual creation of accessibility features in export products?

Every question may not be applicable to each tool, depending on where the tool fits in the workflow process. The image handling workflow can be generally broken down into 3 sections:

1. Image Creation and Editing – direct image editing/processing
2. Image Management – image storage, backend retrieval, search, etc.
3. Content Publishing - final end-user consumable, for example an e-book, etc.

Some applications handle more than one of the workflow sections above. However, in general, questions 1 and 2 apply to image editing tools used in step 1, while questions 3 and 4 apply to document publishing tools used in step 3.

Image Creation and Editing Tools

Image metadata creation often starts at the acquisition phase, inside the camera automatically. Camera setup information is often stored, ambient lighting information is captured, and GPS information is added when such a feature is available and enabled. Depending on which image format is saved, and what the camera capabilities are for metadata information, the image file is written with metadata saved inside the file.

When the pictures are brought back to the production facility, they are typically downloaded to a computer using an image editing application such as Adobe's Photoshop, where the image is cropped, touched up, color adjusted, etc. prior to being archived or otherwise combined with other files to create the final target product, be it Web site, e-book, or other document.

Because of its large market share and widespread use in the publishing industry, the bulk of this section highlights Photoshop's metadata handling although several other image editing tools are also reviewed.

Photoshop

Photoshop has a properties page specific to image metadata, as shown in the screen images below (from their Photoshop version 12 Creative Suite CS5 product).

Figure 1 shows a sample image file into which some description metadata has been inserted (using Photoshop itself.)

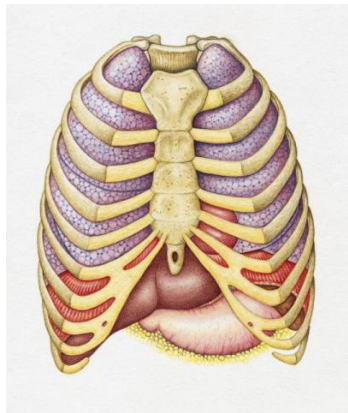


Figure 1 –Sample image File With Embedded Metadata

Figure 2 on the following page shows the "File Info" window for the sample image file. The plethora of metadata fields are divided into logical groupings based on a combination of semantics and metadata format. Notice, for example, that descriptive tags such as author, title, keywords, etc. are grouped into the "Description" tab of the window.

Notice that the “Description” field has some sample information in it. It has been filled with the phrase below in order to track it through the other tabs and potential outputs.

“An illustration of the chest shows lungs positioned behind the rib cage.”

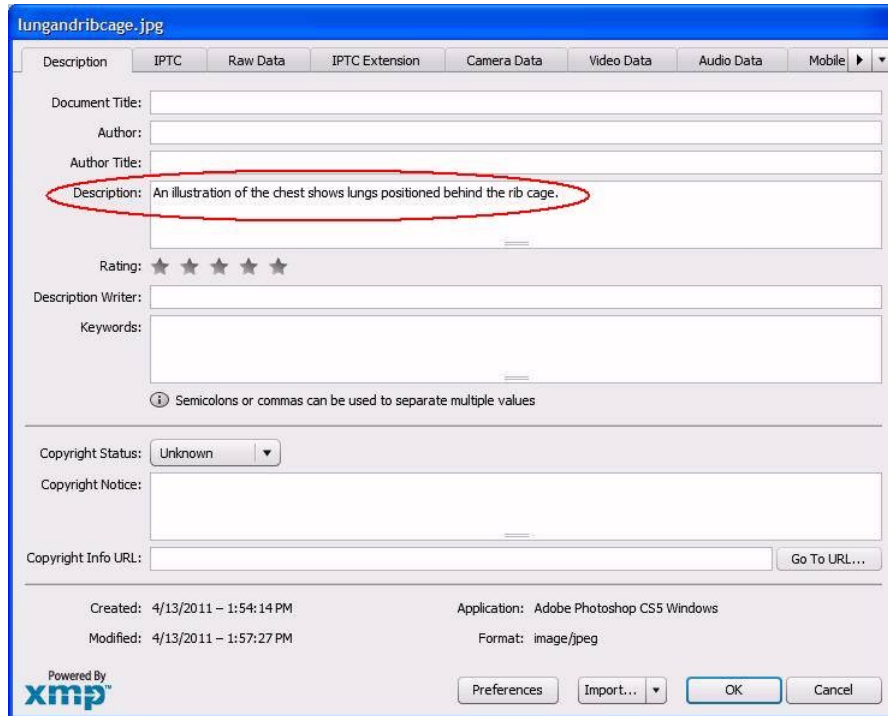


Figure 2 –Description tab in Photoshop FileInfo

Figure 3 on the following page shows the “IPTC” tab of the File Info window with the relevant portion of the IPTC metadata displayed. These are the Core IPTC fields, as distinguished from the IPTC Extension fields. Notice that the data entered in the “Description” field under the “Description” tab is replicated in the “Description” field of this tab. Recall one of the requirements of the Metadata Working Group (see section above):

“It SHOULD keep all forms of metadata it modifies in sync with each other.”

Since Adobe Photoshop is a metadata “Changer” application, it is required (by the Metadata Working Group specification) to synchronize all of the different forms of metadata, resolving and reconciling conflicts according to the semantics. So in this case the application has decided that the IPTC “Description” tag (this corresponds to the “Caption” field of the IPTC Core fields) shares the data that is in the main “Description” field of the FileInfo window.

Figure 4 on the following page shows the “Mobile SWF” tab of the File Info window. It also shares the “Description” field.

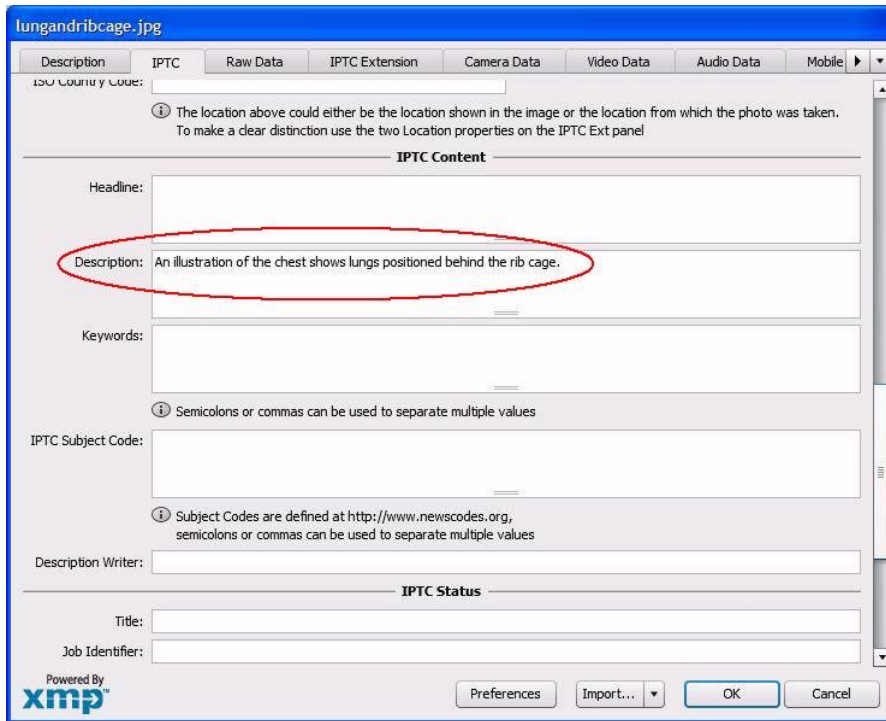


Figure 3 –IPTC tab in Photoshop FileInfo

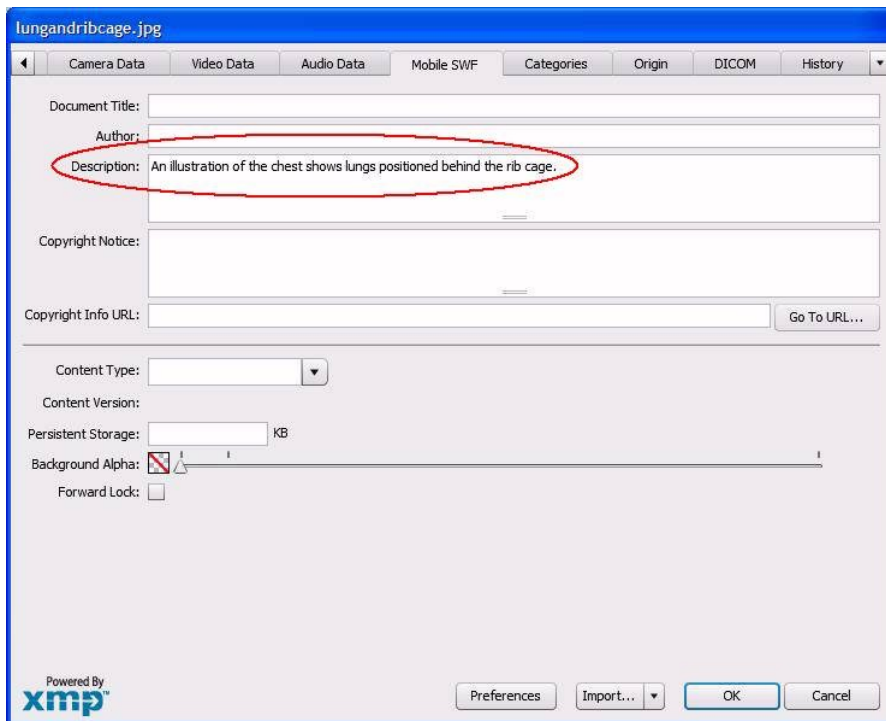


Figure 4 –Mobile SWF tab in Photoshop FileInfo

Figures 5 and 6 respectively show the “Advanced” and “Raw Data” tabs of the File Info window. The Advanced tab (Figure 5) shows the fields in a Property Tree format.

What is notable is that the description data that was originally entered and shared amongst several fields, as seen in Figures 2 through 4, is actually stored internally (to Photoshop) in a single entity. Photoshop engineers have decided that description data most logically resides in the Dublin Core field “dc:description”. It doesn’t appear in any of the other branches of the tree. This allows Photoshop to more easily reconcile and synchronize the various fields, as it is required to do according to the Metadata Working Group specification.

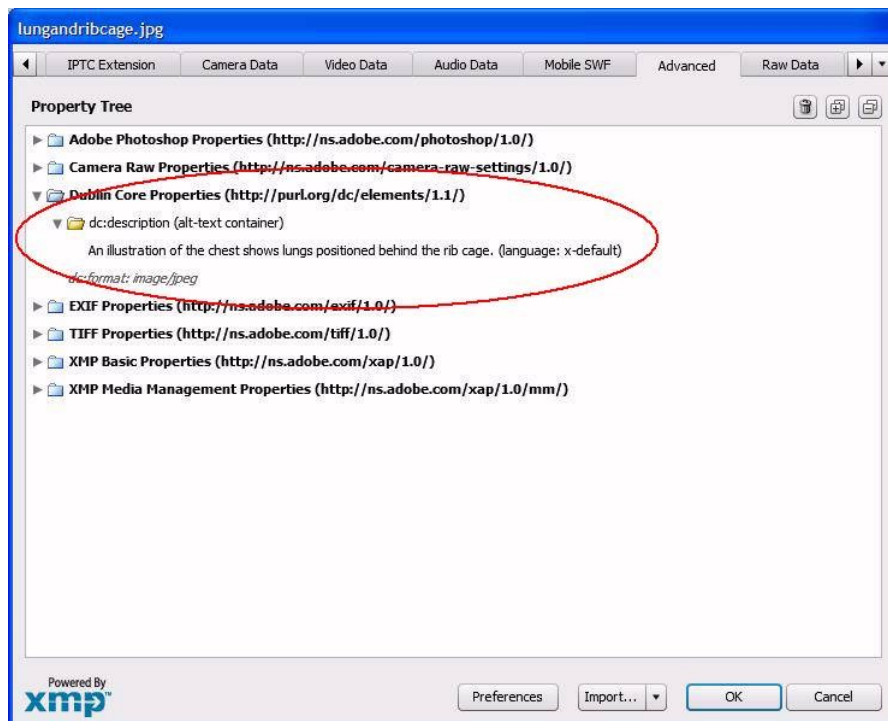


Figure 5 –Advanced tab in Photoshop FileInfo

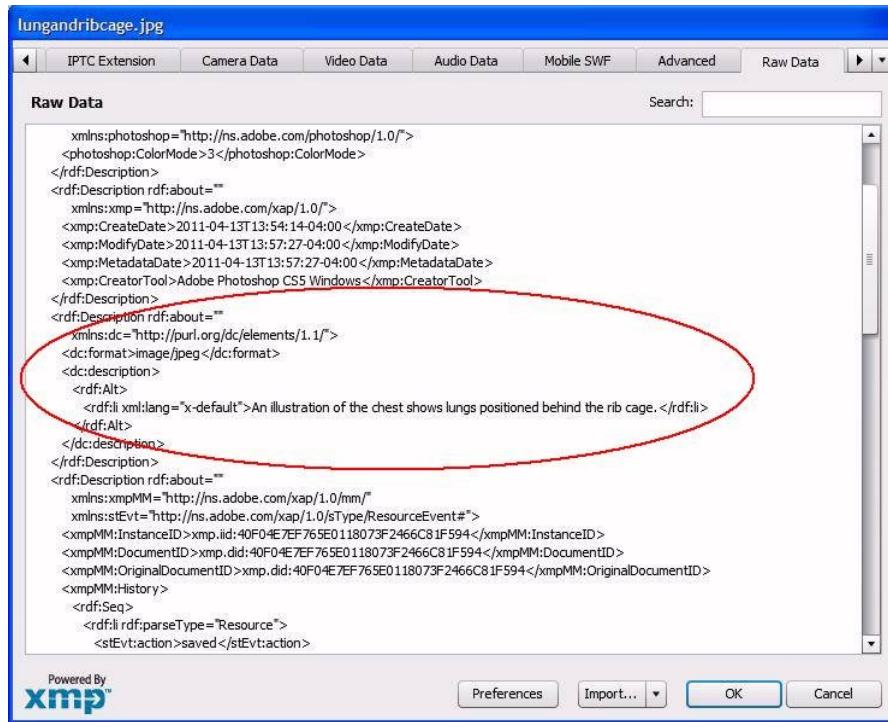


Figure 6 –Raw Data tab in Photoshop FileInfo

Figure 6 shows the raw data as it is stored inside of Photoshop. Note that it is stored in RDF format, which is a W3C specification designed to facilitate data merging, exactly what Photoshop does with potentially disparate metadata fields.

It is clear that Adobe Photoshop conforms (at least in this regard) to the requirements of the Metadata Working Group specification.

While Photoshop is designed to be operated by a human being, Adobe also offers an open source software library, which can be integrated into custom software. From <http://www.adobe.com/devnet/xmp.html>:

The XMP Toolkit allows you to integrate XMP functionality into your product or solution. It supports Macintosh, Windows, as well as UNIX and comes with samples, documentation, source code and project files. The XMP Toolkit is available under the BSD license. The specification is provided under the [XMP Specification Public Patent License](#) (PDF, 24k).

The XMP Toolkit SDK contains two libraries, XMPCore and XMPFiles. XMPCore and XMPFiles are provided as C++ implementations. XMPCore is also provided as a Java version.

This library would theoretically allow a software developer to add image metadata extraction functionality to an application.

To summarize Photoshop's key metadata capabilities:

- Allows Authoring of Embedded Image Metadata: YES
- Preserves Upstream Embedded Image Metadata: YES
- Automatically Populates Downstream Accessibility Features In Exports: N/A
- Allows Manual Creation of Accessibility Features in Export Products: N/A

Other Image Editing Tools

The following are additional tools that have some known effects on metadata, useful for planning workflows around image description.

Illustrator

For the sake of completeness, we have included Adobe's Illustrator product in this analysis. In the publishing workflow, Illustrator is at the same level as Photoshop. Where Photoshop is used for post-processing camera-based images, Illustrator is used to create drawings and vector-based graphical elements.

Illustrator's user interface contains a dialog for viewing metadata that is similar to the one in Photoshop. But it is read-only, does *not* allow authoring of new metadata, and, more importantly, is not aware of upstream image metadata. The "Description" tag authored in the Photoshop example above was not displayed when that image was imported into Illustrator. So Illustrator does not seem to be aware of upstream metadata, and furthermore does not seem to allow for the creation of such metadata in its export products. Authoring of such metadata would have to be accomplished with a different downstream tool.

To summarize Illustrator's key metadata capabilities:

- Allows Authoring of Embedded Image Metadata: NO
- Preserves Upstream Embedded Image Metadata: NO
- Automatically Populates Downstream Accessibility Features In Exports: N/A
- Allows Manual Creation of Accessibility Features in Export Products: N/A

ExifTool

ExifTool is a platform-independent Perl library plus a command-line application for reading, writing and editing metadata information in a wide variety of files. ExifTool supports many different metadata formats including EXIF, GPS, IPTC, XMP, JFIF, GeoTIFF, ICC Profile, Photoshop IRB, FlashPix, AFCP and ID3.

To summarize ExifTool's key metadata capabilities:

- Allows Authoring of Embedded Image Metadata: YES
- Preserves Upstream Embedded Image Metadata: YES*
- Automatically Populates Downstream Accessibility Features In Exports: N/A
- Allows Manual Creation of Accessibility Features in Export Products: N/A

* Indicated from web and other documentation, not direct testing

See: <http://www.sno.phy.queensu.ca/~phil/exiftool/>

ImageMagick

ImageMagick is a Windows-only software suite that can create, edit, compose, or convert bitmap images. It is most often used as a command-line tool for batch processing of images. While it does not offer any metadata *authoring* capabilities, tests with ImageMagick indicate that existing metadata is preserved during processing.

To summarize ImageMagick's key metadata capabilities:

- Allows Authoring of Embedded Image Metadata: NO
- Preserves Upstream Embedded Image Metadata: YES
- Automatically Populates Downstream Accessibility Features In Exports: N/A
- Allows Manual Creation of Accessibility Features in Export Products: N/A
-

See: <http://www.imagemagick.org/script/index.php>

GIMP

GNU Image Manipulation Program (GIMP) is freely distributed open source software for photo retouching, image composition and image authoring. It is in wide use on Linux and other *nix operating systems. The public discussion lists indicate that GIMP does not preserve IPTC data or XMP data, but will conserve (unchanged) EXIF data if you compile against libexif and choose that option (which is selected by default) in the jpeg save dialog.

To summarize GIMP's key metadata capabilities:

- Allows Authoring of Embedded Image Metadata: NO
- Preserves Upstream Embedded Image Metadata: YES**
- Automatically Populates Downstream Accessibility Features In Exports: N/A
- Allows Manual Creation of Accessibility Features in Export Products: N/A
-

See: <http://www.gimp.org/>

<http://www.groupsrv.com/computers/about22938.html>

Image Management Tools

Image management tools are designed to help organize large sets of images. They are typically used by professionals for image processing and backend storage and retrieval. Some have extensive metadata integration for searching, etc., as described below.

LightRoom (Adobe)

Lightroom, from Adobe, is an application designed to manage large sets of images that allows the user to create, manage, and search on embedded metadata. Lightroom is a standalone application

*** Some, but limited

for photo collection management, but many of the same metadata search and retrieval features are available in their Bridge product described below.

To summarize LightRoom's key metadata capabilities:

- Allows Authoring of Embedded Image Metadata: YES
- Preserves Upstream Embedded Image Metadata: YES*
- Automatically Populates Downstream Accessibility Features In Exports: N/A
- Allows Manual Creation of Accessibility Features in Export Products: N/A

Bridge (Adobe)

Adobe's Bridge is included in Adobe's Creative Suite which bundles PhotoShop, Illustrator, InDesign and other tools. Bridge has many of the same metadata search and retrieval capabilities as its standalone cousin, Lightroom.

To summarize Bridge's key metadata capabilities:

- Allows Authoring of Embedded Image Metadata: YES
- Preserves Upstream Embedded Image Metadata: YES
- Automatically Populates Downstream Accessibility Features In Exports: N/A
- Allows Manual Creation of Accessibility Features in Export Products: N/A

interMedia (Oracle)

Oracle is a leading database software provider and has a product called interMedia, which has image metadata search and retrieval capabilities. For back-end CMS systems that employ Oracle databases, the interMedia features could allow a website to extract image description metadata and include it in the HTML output.

See: http://download.oracle.com/docs/cd/B19306_01/appdev.102/b14302/ch_intr.htm
http://download.oracle.com/docs/cd/B19306_01/appdev.102/b14259/xdb_repos_meta.htm
[- sthref2326\)](#)

Oracle interMedia ("interMedia") is a feature that enables Oracle Database to store, manage, and retrieve images, audio, video, or other heterogeneous media data in an integrated fashion with other enterprise information. Oracle interMedia extends Oracle Database reliability, availability, and data management to multimedia content in traditional, Internet, electronic commerce, and media-rich applications. Oracle interMedia does not control media capture or output devices; this function is left to application software. interMedia manages multimedia content by providing the following:

- Storage and retrieval (see [Section 1.7.1](#), [Section 1.8](#), and [Section 1.9](#))
- Media and application metadata management (see [Section 1.3](#), [Section 1.4](#), [Section 1.5](#), and [Section 1.6](#))
- Support for popular formats (see the audio, image, and video data format appendixes in [Oracle interMedia Reference](#))
- Access through traditional and Web interfaces (see [Section 1.9](#))
- Querying using associated relational data (see [Section 1.7](#))
- Querying using extracted metadata (see [Section 1.5.4](#))
- Querying using media content with optional specialized indexing (see [Section 1.7.2](#))

* Indicated from web and other documentation, not direct testing

interMedia provides media content services to Oracle JDeveloper, Oracle Content Management SDK, Oracle Application Server Portal, and Oracle partners. This guide describes the management and integration of audio, image, and video, or other heterogeneous media data with other Oracle tools and software, as well as with third-party tools and software.

To summarize interMedia's key metadata capabilities:

- Allows Authoring of Embedded Image Metadata: YES*
- Preserves Upstream Embedded Image Metadata: YES*
- Automatically Populates Downstream Accessibility Features In Exports: N/A
- Allows Manual Creation of Accessibility Features in Export Products: N/A

Content Publishing Tools

There is no single set of tools, nor single workflow, in the electronic publishing industry. Each publisher, from small to large, selects from an array of available software tools. Small entities might combine open source tools with tools of moderate cost. Large enterprise operations often have enterprise-level, CMS-based back ends, but combine them with some of the same editing and exporting tools as small operations. There is a lot of mixing and matching.

The specific tools that are used at any given step in the workflow are typically a function of history and comfort levels. Adobe InDesign has a large market share, but many people and companies also use QuarkXpress. And while larger publishing companies are trying to move toward using federated and automated content management systems, the actual manipulation of individual files is still handled at the manual level using single-user tools such as InDesign, QuarkXpress, and open source editing and batch processing tools.

The following is discussion of some currently popular downstream applications relative to metadata integration workflow.

Dreamweaver (Adobe)

While website publishing is not of direct interest in this discussion, it is instructive to look at this very common process. This is because HTML already includes well-understood mechanisms for inclusion of image description metadata, the well-known *@alt* of the ** element.

Recall that an image can have a Description metadata tag embedded into it using Adobe Photoshop. The likeliest downstream application to utilize image metadata authored in (for example) Adobe Photoshop would be Adobe Dreamweaver, their application for developing websites in HTML and other supported languages and technologies.

When you insert an image into a webpage using Dreamweaver, the “alt” attribute is *not* automatically filled in using the existing metadata in the image. Dreamweaver *does*, of course, allow you to set the *@alt* attribute manually, via a dedicated window, but it is not *automatically* filled in.

** Indicated from web and other documentation, not direct testing

Since Adobe (for example) already has the logic and codebase required to extract image metadata, it is likely a relatively easy feature to add an “auto-fill” for @alt text. The user could then be prompted either to leave it as is or edit it, as appropriate.

To summarize Dreamweaver’s key metadata capabilities:

- Allows Authoring of Embedded Image Metadata: N/A
- Preserves Upstream Embedded Image Metadata: N/A
- Automatically Populates Downstream Accessibility Features In Exports: NO
- Allows Manual Creation of Accessibility Features in Export Products: YES

InDesign (Adobe)

InDesign from Adobe is a popular layout and publishing application. Inputs to InDesign are typically graphics (for example photos from Photoshop or vector art from Illustrator) and text. Outputs (export products) can include Dreamweaver, EPUB, PDF, and various XML formats.

InDesign has a mechanism called “Structure”, for tagging elements in a project. For an image element, which is known as a “Figure”, you can add items to the structure tree for that Figure which InDesign understands to be alternative text. If, for example, a Figure or Image is given an “Alt” sub-tag, along with some descriptive text, that data will be mapped into the corresponding elements during PDF (alternative text) output, such that a screen reader will be able to voice the appropriate descriptive text. InDesign does not currently utilize existing embedded image metadata in this process.

Unfortunately, the same mapping is *not* done for InDesign’s Dreamweaver / HTML output. Corresponding HTML output sets the default @alt attribute of the element to the *filename* of the image. Web accessibility guidelines explicitly suggest *useful* information or *null* value for @alt, rather than a filename. (This is because a filename will be voiced, adding to useless audio overload, whereas a null value will simply be silent.)

As noted above, Dreamweaver itself does not utilize existing embedded description metadata for populating its internal @alt settings dialogs. This represents another workflow intersection and point of interest.

Of particular note for the potential processes that utilize embedded image metadata, when creating Dreamweaver and EPUB output products in InDesign, embedded image metadata will only be preserved if the user selects “Original” or “Link to server path” on “Images” tab of the intervening XHTML Export Options dialog. The initial default selection for that setting is “Optimized”, which removes the embedded metadata. Fortunately, the user’s setting for that item is remembered between sessions. Thus, InDesign can be said to “preserve” existing embedded image metadata, although it is not really modifying the image, but simply making a simple copy of the image file.

To summarize InDesign's key metadata capabilities:

- Allows Authoring of Embedded Image Metadata: N/A
- Preserves Upstream Embedded Image Metadata: YES**
- Automatically Populates Downstream Accessibility Features In Exports: NO
- Allows Manual Creation of Accessibility Features in Export Products: YES

Acrobat (Adobe)

As noted above, PDF output from InDesign, if carefully crafted in InDesign in the first place, will contain metadata sufficient to make the PDF document accessible via screen readers. Additionally, Acrobat contains a user interface for creating that accessibility metadata manually. It will not currently utilize existing embedded image metadata.

To summarize Acrobat's key metadata capabilities:

- Allows Authoring of Embedded Image Metadata: N/A
- Preserves Upstream Embedded Image Metadata: N/A
- Automatically Populates Downstream Accessibility Features In Exports: NO
- Allows Manual Creation of Accessibility Features in Export Products: YES

QuarkXpress (Quark)

QuarkXpress is a print design and publishing tool used by publishers. It has some abilities to utilize various inputs from the Adobe Creative Suite as well as some other XML formats, and can also output to some of those formats, thereby allowing it to exist comfortably in heterogeneous work flows. The documentation for QuarkXpress v.8 contains no information about accessibility, or tagging for such. The assumption is that its output is not directly accessible. The output products can, of course, be made accessible in other tools, as described elsewhere in this paper.

To summarize Quark's key metadata capabilities:

- Allows Authoring of Embedded Image Metadata: N/A
- Preserves Upstream Embedded Image Metadata: N/A
- Automatically Populates Downstream Accessibility Features In Exports: NO*
- Allows Manual Creation of Accessibility Features in Export Products: ?? (unknown)

Table 4 on the following page summarizes the accessibility features of each product relative to image metadata.

Table 4 — Product Features Relative to Embedded Image Metadata

Image Editing Tools

<i>Software Tool</i>	<i>Allows Authoring of Embedded Image Metadata</i>	<i>Preserves Upstream Embedded Image Metadata</i>	<i>Automatically Uses Upstream Image Metadata to Populate Accessibility Dialogs for Export Products</i>	<i>Allows Manual Authoring of Accessibility Data in Export Products</i>
Photoshop (Adobe)	YES	YES	--	--
Illustrator (Adobe)	NO	NO	--	--
ExifTool	YES	YES*	--	--
ImageMagick	NO	YES	--	--
Gimp	NO	YES**	--	--

Image Management Tools

<i>Software Tool</i>	<i>Allows Authoring of Embedded Image Metadata</i>	<i>Preserves Upstream Embedded Image Metadata</i>	<i>Automatically Uses Upstream Image Metadata to Populate Accessibility Dialogs for Export Products</i>	<i>Allows Manual Authoring of Accessibility Data in Export Products</i>
Lightroom (Adobe)	YES	YES*	--	--
Bridge (Adobe)	YES	YES	--	--
interMedia (Oracle)	YES*	YES*	--	--

Content Publishing Tools

<i>Software Tool</i>	<i>Allows Authoring of Embedded Image Metadata</i>	<i>Preserves Upstream Embedded Image Metadata</i>	<i>Automatically Uses Upstream Image Metadata to Populate Accessibility Dialogs for Export Products</i>	<i>Allows Manual Authoring of Accessibility Data in Export Products</i>
Dreamweaver (Adobe)	--	--	NO	YES
InDesign (Adobe)	--	YES**	NO	YES
Acrobat (Adobe)	--	--	NO	YES
QuarkXpress	--	--	NO*	??

* indicated from web and other documentation, not direct testing

** Some, but limited

?? Not yet evaluated

-- Indicates not applicable or application is not used to create/modify images

Summary

This analysis identifies current opportunities and challenges in development of solutions that both allow original authors to create descriptions that can survive the publishing workflow and allow subject matter experts or others to modify or add contextually appropriate descriptions at different points along the production and distribution chain. Both of these capabilities are essential to the creation of high-quality accessible digital books.

As outlined in this paper, there are several metadata standards in wide use today for embedding information into digital images. Each of the identified metadata specifications contains one or more fields that could be used to carry image description embedded directly in the image file. The definitions of these fields in their respective specifications indicate they should be used for descriptive information about the image. While those standards were and are continuing to develop independently, there are efforts to create a rational framework for integration of metadata stored in those disparate formats.

Equally important, while some tools specifically handle embedded image metadata, many do not. The typical publication workflow utilizes many different tools, and it is not currently possible to maintain embedded metadata all the way through the chain. However, there is enough of a germ of metadata handling in several of the existing tools that there are good opportunities for discussions with the tool developers to extend the capabilities across a wider section of their product lines. There are discussions currently underway with some of those developers to explore these opportunities.

Useful References

<http://exif.org/specifications.html>

<http://exif.org/Exif2-1.PDF>

<http://exif.org/dcf.PDF>

<http://www.jeita.or.jp/english/>

<http://www.iptc.org/site/Home/>

http://www.iptc.org/std/photometadata/specification/IPTC-PhotoMetadata-201007_1.pdf

<http://dublincore.org/documents/dces/>

<http://www.metadataworkinggroup.org/specs/>

http://www.metadataworkinggroup.org/pdf/mwg_guidance.pdf

<http://www.adobe.com/devnet/xmp.html>

<http://www.w3.org/TR/SVG-access/>

<http://www.w3.org/TR/SVG/struct.html - DescriptionAndTitleElements>

http://www.w3schools.com/svg/svg_rect.asp

<http://www.daisy.org/z3986/2005/Z3986-2005.html#PubMed>

<http://manuals.viewplus.com/en/view/7/55/>