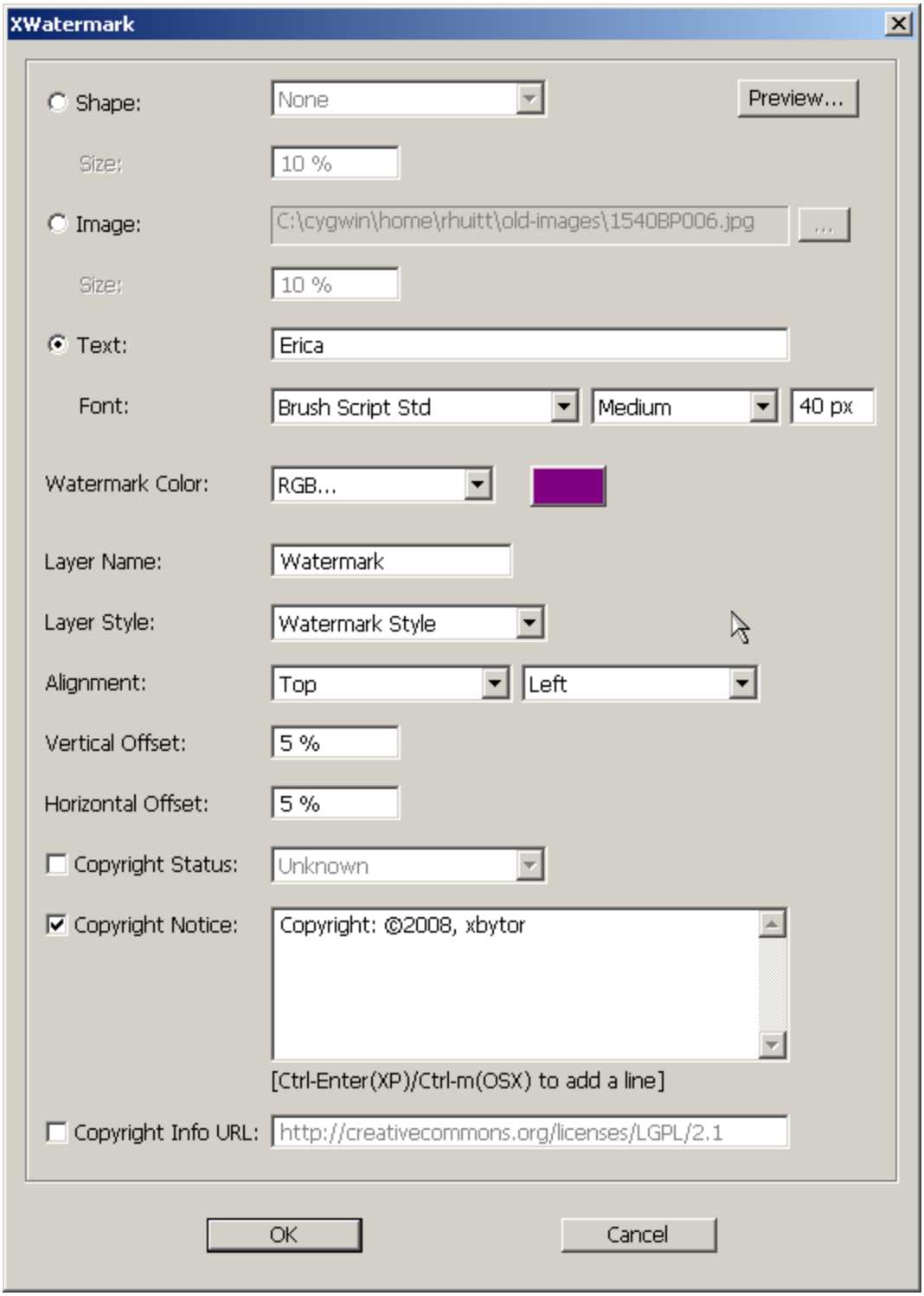

XWatermark.jsx is a Photoshop JavaScript for adding watermarks and copyright information to images. It can be used as a standalone script, as part of a batch process, or embedded into a large image processing script.

The XWatermark UI

The XWatermark user interface lets a user specify what to use for a watermark (some text, a shape, or an image), attributes about the watermark (its color and style), and the location of the watermark within the image. Copyright information may also be specified at the same time.

Watermark Type

Watermarks can either be a Custom Shape, an image file, or text. The *Shape*, *Image*, and *Text* radio buttons select between these three different types of watermarks.

Watermark Shape

The menu provided for selecting a watermark shape contains the current set of Custom Shapes available in Photoshop. The '*None*' shape means that no shape will be inserted. This can be used as a placeholder or when using the script to just set copyright information. The '*Load Shape...*' item lets a user load a shape from a Custom Shape file that was previously saved in Photoshop. The only restriction that the script places on this file is that there is only one shape in the *.csh* (custom shape) file.

The 'Size' property is used to specify the size of the largest dimension of the shape when it is placed into the image. By default, Size is expressed in pixels, but may also be in other units. If the size is specified as a percent (e.g. “10%”), the maximum dimension of the shape will be that percentage of the largest dimension of the image. For instance, if the image is 3500x2000 pixels and the shape size is set to 10%, the size of the shape will fit an area of 350x350 pixels. See the *Units* appendix for more information.

Watermark Image

Existing image files may be used for watermarks. While png and gif files typically work best because of their transparency capabilities, most Photoshop-compatible image formats are supported.

The 'Size' property is used to specify the size of the largest dimension of the watermark when it is placed into the containing image. By default, Size is expressed in pixels, but may also be in other units. If the size is specified as a percent (e.g. “10%”), the maximum dimension of the watermark will be that percentage of the largest dimension of the containing image. For instance, if the image is 3500x2000 pixels and the watermark size is set to 10%, the size of the watermark will fit an area of 350x350 pixels. See the *Units* appendix for more information.

Watermark Text

A watermark may also simply be a bit of text. This is where that text is specified as well as the font to be used for the text.

The size of the font is a bit special. By default, the font size is expressed in points, but may also be in other units. If the size is specified as a percent (e.g. “10%”), the size of the font will be that percentage of the largest dimension of the image. For instance, if the image is 3500x2000 pixels and the font size is set to 10%, the font size is set to 350 pixels. See the *Units* appendix for more information.

Watermark Attributes

The ***Color*** of the watermark can be specified. This has no effect on image watermarks.

The ***Layer Style*** is the style that is applied to the layer containing the watermark. The list of names in the menu is the set of named styles currently defined in Photoshop. The '*None*' style means that no style will be applied to the layer. The '*Load Style...*' item lets a user load a style from a styles file that was previously saved in Photoshop. The only restriction that the script places on this file is that there is only one style in the *.asl* (styles) file. The '*DefineStyle...*' item allows a user to dynamically define a style by opening up a layer styles editor on a sample document. Unless a dynamically defined style is saved as a part of the editing process, it will no longer exist when the script has finished. Style definitions are not saved in the script's settings file.

Watermark Location

The placement of the watermark in the image is determined by a set of location properties. The vertical and horizontal ***Alignment*** properties indicate which edges a watermark should be placed relative to. The vertical and horizontal ***Offset*** properties are used to specify the how far from the edges the watermark should be placed. By default, ***Offsets*** are expressed in pixels, but may also be in other units. If the horizontal is specified as a percent (e.g. “10%”), the location of the watermark will be inset that amount horizontally. For instance, if the image is 3500x2000 pixels and the horizontal offset is set to 10% and is aligned left, the left edge of the watermark 350 pixels inset from the left edge of the image. When both the horizontal and vertical offsets are expressed as identical percentages (e.g. 5%), the pixel offset is computed for both and the larger of the two is used for both offsets. This allows the watermark to be the same number of pixels from both edges without having to specify exactly how many pixels that should be.

Copyright Information

This script also lets a user optionally specify copyright information that will be added to the metadata of processed images.

Appendix A: Units

Units

Several fields in the XWatermark UI are *Unit* fields. That is, they define the size of something by use of a value and a type for that value. For instance, a Size field my be specified as “25 px”, “2.5 in”, or “20%”. The 25, 2.5, and, 20 parts are the *values* of these fields while the “px”, “in”, and “%” portions are the *types* of those fields.

The following set of types are recognized:

Type	*Abbreviation*	*Note*
Inches	in	2.54 cm
Feet	ft	30.48 cm
Yards	yd	91.44 cm
Miles	mi	1609.344 m
Millimeters	mm	
Centimeters	cm	
Meters	m	
Kilometers	km	
Points	pt	inches / 72
Picas	pc	points * 12
Traditional Point	tpt	inches / 72.27
Traditional Pica	tpc	12 tpt
Ciceros	ci	12.7872 pt
Pixels	px	
Percent	%	

The abbreviation of the type is used in the UI field.

When '%' is used as the type, the largest dimension of the underlying object is used to compute the resulting value for the field. For instance, if the underlying object is 1024x500 pixels in size and the field is set to “25 %”, the computed value for the setting will be 256 pixels.

The font field defaults to *pt* as its type. All others default to *px*. If no type is specified in the field, the default type for that field is assumed.

Appendix B: Metadata Text Substitution

Watermark text may optionally include information from the image's metadata. This is accomplished by inserting *format specifiers* as a part of the text in the Text field. For instance, setting the field to 'NYC: %C' could result in a watermark with the text 'NYC: 2008-04-01'. All EXIF, IPTC, and XMP metadata is available using this mechanism. It is also possible to alter the formatting of date, time, and GPS metadata fields.

Format Specifiers

A format string has text in it along with 'special' characters (aka *patterns*) that are substituted with additional information about an image, such as EXIF metadata. These substitution patterns always begin with a % character which is followed by one or more characters. For instance, *%W* and *%H* are the pattern specifiers for the width and height of the image, respectively. A format specifier of "*%Wx%H*" would result in text that looked something like "*640x480*".

This table lists all of the caption format specifiers and their substitutions.

Specifier	**Substitution**
%%	%
%d	Directory (full path)
%e	Extension
%f	Filename without Extension
%p	Parent Directory (name only)
%s	Size of File
%S	Size of File using K/M/G shorthand
%B	Bits Per Channel
%C	File Creation Date
%C{dateFormat}	File Creation Date using specified format
%E{tag}	EXIF field
%F	Filename (same as %f.%e)
%H	Image Height (pixels)
%I{tag}	IPTC file (case insensitive)
%M	Color Mode
%N	Current Date
%N{dateFormat}	Current Date using specified format
%P	Color Profile
%R	Resolution (ppi)
%T	File Modification Date
%T{dateFormat}	File Modification Date using specified format
%W	Image Width (pixels)
%X{space:tag}	XMP Field from specified space
%X{tag}	XMP Field (from first space with tag)

These examples illustrate how basic substitutions occur:

Format	*Output*
%Wx%H %Rppi	640x480 72ppi
%f.%e	bigFish.jpg
%P %M	sRGB IEC61966-2.1 Lab

Named Specifiers

In addition to the basic specifiers, there are the *named* specifiers. Because of the very large numbers of possible metadata fields, it is impractical to use simple one letter specifiers for them all. Because of this, the names of EXIF, IPTC, and XMP metadata fields can be used in format specifiers. The general form for a EXIF specifier would be *%E{tag}* where the *%E* indicates that this is an EXIF specifier and *tag* is the name of the EXIF field that we want. For instance *%E{Model}* specifies that we want the *Model* EXIF field, which would be substituted by something like *NIKON D200*.

EXIF metadata is image file information typically created by a digital camera (or some other image file creator). The information available varies depending on the creator of the image; one vendor may provide basic fields like *aperture* and *ISO speed* information while another may provide those fields and *GPS* data.

Here is a partial list of EXIF fields that might be found in an image
Aperture Value, Artist, Color Space, Date Time, Date Time Original, Exposure Bias Value, Exposure Mode, Exposure Program, Exposure Time, F-Stop, Flash, Focal Length, ISO Speed, Ratings, Make, Max Aperture Value, Metering Mode, Model, Orientation, Software, Shutter Speed, White Balance.

These examples illustrate how EXIF substitutions occur:

Format	*Output*
%E{F-Stop} %E{Exposure Time}	f/5.0 2.5 sec
ISO%E{ISO Speed Ratings}	ISO320

IPTC metadata is image file information that was originally developed by news organizations and, later, by Adobe. This specifies information like *Title*, *Author*, and *Keywords*.

Here is a partial list of IPTC fields that may be found in an image.
Author, AuthorPosition, Caption, CaptionWriter, Category, City, Copyrighted, CopyrightNotice, Country, CreationDate, Credit, Headline, Instructions, JobName, Keywords, OwnerURL, ProvinceState, Source, SupplementalCategories, Title, TransmissionReference, Urgency.

These examples illustrate how IPTC substitutions occur:

Format	*Output*
%I{Keywords}	Animals, Kangaroo

%I{ProvinceState}, %I{Country}	Queensland, Australia

The remaining category of *named* fields is XMP. XMP is an open-ended XML-based format in which all sorts of information may be stored. XMP metadata is subdivided into namespaces which group fields into collections. There is, for instance, a namespace *xapRights* the specifies rights management (copyright) information for an image. You typically specify the namespace and the field/property within that name space that you want. For instance *%X{xapRights:Copyright}* would be substituted with the *Copyright* property from the *xapRights* namespace and would look something like *©2008 xbytor*.

These examples illustrate how IPTC substitutions occur:

Format	*Output*
%X{dc:format}	image/tiff
%X{DocumentID}	uuid:76AC19F1C0EFDB11A4298F44C792E00A

You can, of course, mix the different kinds of format specifiers like this:

Format	*Output*
%X{format} %S - "%I{Title}"	image/tiff 987K - "Mardi Gras 2007"

Date Formats

There are three basic specifiers that map to dates *(%C, %N, %T*). When the substitution occurs for these specifiers, the dates have to be formatted somehow. By default, in CSX, the dates are formatted *YYYY-MM-DD*. *%C* would be substituted by something that looks like *2007-03-21*. The default date format for CSX is specified on the *Advanced* panel in the **dateFormat** setting. Changing this effects the formatting of dates throughout CSX. You can, however, override the default date format by specifying the desired format as part of the specifier. For instance, you could specify a file creation date like this *%C{%m/%d/%Y}* or (equivalently) like this *%C{MM/DD/YYYY}*. See the ***Date Format Specification*** appendix for details on date formats. There are also EXIF, IPTC, and XMP date fields that we have to worry about. As a general rule if a *named specifier* contains '*date*' anywhere in its name, CSX will attempt to format it using the default date format.

GPS Formats

CSX provides a mechanism for specifying the format of GPS Coordinates. On the *Advanced* panel there is a **gpsFormat** setting. Much like the **dateFormat** setting, it allows a format for GPS data to be specified. Instead of inventing some again, I duplicated the format used by *exiftool* (http://www.sno.phy.queensu.ca/~phil/exiftool/).

The format for GPS Coordinates uses a syntax similar to other formats in CSX. The specifiers in the format correspond to degrees, minutes, and seconds in that order with the minutes and seconds being optional. In this format, *%d* means the value should be formatted as an integer while a *%f* means the value should be formatted as a floating point number. For precise details on alignment, padding, etc... , see http://www.opengroup.org/onlinepubs/007908799/xsh/fprintf.html.

These examples give the output of the same coordinate using various formats:

Format	*Output*
%d deg %d' %.2f"	54 deg 59' 22.80" (the default)
%d deg %.4f min	54 deg 59.3800 min
%.6f degrees	54.989667 degrees

GPS formatting only occurs for the EXIF fields *GPS Longitude* and *GPS Latitude*.

Appendix C: Date Format Specification

Date formats can be specified using one of two styles.

The first style of format specification has these mappings:

YYYY	Four digit year. *2006*
YY	Two digit year. *06*
MM	Two digit month. *10*
DD	Two digit month. *31*
H	Two digit hour on a 24 hour clock. *20*
I	Two digit hour on a 12 hour clock. *08*
M	Two digit minute. *43*
S	Two digit seconds. *02*
P	AM or PM. *PM*

Using these formats, you could create a text layer with the name *@Date*, set its contents to *"Created on YYYY/MM/DD"*, and end up with a layer that looks like "Created on 2006/10/31".

The second style of format specification is based on the Unix *strftime(3)* implementation. The format specification is text that includes characters that will be substituted according to the following rules. Characters in a format specification that are not matched are left unchanged. CSX determines which specification to use by looking for a '%' character in the format specification. If the format does **not** contain a '%', the Y/M/D-style substitution is used. If it does, this style is used.

A date/time of October 31, 2006 20:43:02 will be used for the example below.

Specifier	**Substitution**
%a	A three-letter abbreviation for the day of the week, one of 'Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', or 'Sat'. *Tue*
%A	The full name for the day of the week, one of 'Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', or 'Saturday'. *Tuesday*
%b	A three-letter abbreviation for the name of the month, one of 'Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', or 'Dec'. *Oct*
%B	The full name of the month. One of 'January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', or 'December'. *October*
%c	The complete date and time in this format:

	"Tue Oct 31 20:43:02 2006".
%C	The century portion of the year, zero padded. *20*
%d	The day of the month, zero padded. *31*
%D	The date in the format *"%m/%d/%y"*. *10/31/06*
%e	The day of the month, space padded. *31*
%F	The date in the format *"%Y-%m-%d"*. *2006-10-31*
%h	A three-letter abbreviation for the name of the month, same as *%b*. *Oct*
%H	The hour on a 24 hour clock, zero padded. *20*
%I	The hour on a 12 hour clock, zero padded. *08*
%j	The Julian date. *304*
%k	The hour on a 24 hour clock, space padded. *20*
%l	The hour on a 12 hour clock, space padded. *8*
%m	The month number, zero padded. *10*
%M	The minutes, zero padded. *43*
%n	A newline character.
%p	Either AM or PM as appropriate. *PM*
%r	12 hour time to the second with this format: "%I:%M:%S %p". *08:43:02 PM*
%S	The seconds, zero padded. *02*
%t	A tab character.
%T	24 hour time to the seconds with this format: *"%H:%M:%S"*. *20:43:02*
%u	The weekday as a number where Monday is 1 and Sunday is 7. *2*

%w	The weekday as a number where Sunday is 0 and Saturday is 6. *2*
%x	The date in the format *"%m/%d/%y"*. Same as *%D*.
%X	24 hour time to the seconds with this format: *"%H:%M:%S"*. Same as *%T*.
%y	The last two digits of the year. *06*
%Y	The full four digits of the year. *2006*
%%	The '%' character.

Using these formats, you could create a text layer with the name *@Date*, set its contents to *"Created on %Y/%m/%d"*, and end up with a layer that looks like "Created on 2006/10/31".