NAME

exiftool - Read and write meta information in files

SYNOPSIS

Reading

exiftool [OPTIONS] [-TAG...] [--TAG...] FILE...

Writing

exiftool [OPTIONS] -TAG[+-<]=[VALUE]... FILE...

Copying

exiftool [OPTIONS] -tagsFromFile SRCFILE [-[DSTTAG<]SRCTAG...] FILE...

Other

exiftool [-ver | -list[w|f|r|wf|g[NUM]|d|x]]

For specific examples, see the EXAMPLES sections below.

This documentation is displayed if exiftool is run without an input FILE when one is expected.

DESCRIPTION

A command-line interface to Image::ExifTool, used for reading and writing meta information in a variety of file types. FILE is one or more source file names, directory names, or "-" for the standard input. Metadata is read from source files and printed in readable form to the console (or written to output text files with **-w**).

To write or delete metadata, tag values are assigned using -TAG=[VALUE], and/or the **-geotag**, **-csv=** or **-json=** options. To copy or move metadata, the **-tagsFromFile** feature is used. By default the original files are preserved with "_original" appended to their names -- be sure to verify that the new files are OK before erasing the originals. Once in write mode, exiftool will ignore any read-specific options.

Note: If FILE is a directory name then only supported file types in the directory are processed (in write mode only writable types are processed). However, files may be specified by name, or the **-ext** option may be used to force processing of files with any extension. Hidden files in the directory are also processed. Adding the **-r** option causes subdirectories to be processed recursively, but subdirectories with names beginning with "." are skipped unless **-r.** is used.

Below is a list of file types and meta information formats currently supported by ExifTool (r = read, w = write, c = create):

File Types

360	r/w	DPX	r	ITC	r	NRW	r/w	RAM	r
3FR	r	DR4	r/w/c	J2C	r	NUMBERS	r	RAR	r
3G2	r/w	DSS	r	JNG	r/w	O	r	RAW	r/w
3GP	r/w	DV	r	JP2	r/w	ODP	r	RIFF	r
A	r	DVB	r/w	JPEG	r/w	ODS	r	RSRC	r
AA	r	DVR-MS	r	JSON	r	ODT	r	RTF	r
AAE	r	DYLIB	r	JXL	r	OFR	r	RW2	r/w
AAX	r/w	EIP	r	K25	r	OGG	r	RWL	r/w
ACR	r	EPS	r/w	KDC	r	OGV	r	RWZ	r
AFM	r	EPUB	r	KEY	r	ONP	r	RM	r
AI	r/w	ERF	r/w	LA	r	OPUS	r	SEQ	r
AIFF	r	EXE	r	LFP	r	ORF	r/w	SKETCH	r
APE	r	EXIF	r/w/c	LIF	r	ORI	r/w	SO	r
ARQ	r/w	EXR	r	LNK	r	OTF	r	SR2	r/w
ARW	r/w	EXV	r/w/c	LRV	r/w	PAC	r	SRF	r
ASF	r	F4A/V	r/w	M2TS	r	PAGES	r	SRW	r/w
AVI	r	FFF	r/w	M4A/V	r/w	PBM	r/w	SVG	r

AVIF	r/w	FITS	r	MACOS	r	PCD	r	SWF	r
AZW	r	FLA	r	MAX	r	PCX	r	THM	r/w
BMP	r	FLAC	r	MEF	r/w	PDB	r	TIFF	r/w
BPG	r	FLIF	r/w	MIE	r/w/c	PDF	r/w	TORRENT	r
BTF	r	FLV	r	MIFF	r	PEF	r/w	TTC	r
CHM	r	FPF	r	MKA	r	PFA	r	TTF	r
COS	r	FPX	r	MKS	r	PFB	r	TXT	r
CR2	r/w	GIF	r/w	MKV	r	PFM	r	VCF	r
CR3	r/w	GPR	r/w	MNG	r/w	PGF	r	VNT	r
CRM	r/w	GZ	r	MOBI	r	PGM	r/w	VRD	r/w/c
CRW	r/w	HDP	r/w	MODD	r	PLIST	r	VSD	r
CS1	r/w	HDR	r	MOI	r	PICT	r	WAV	r
CSV	r	HEIC	r/w	MOS	r/w	PMP	r	WDP	r/w
CUR	r	HEIF	r/w	MOV	r/w	PNG	r/w	WEBP	r/w
CZI	r	HTML	r	MP3	r	PPM	r/w	WEBM	r
DCM	r	ICC	r/w/c	MP4	r/w	PPT	r	WMA	r
DCP	r/w	ICO	r	MPC	r	PPTX	r	WMV	r
DCR	r	ICS	r	MPG	r	PS	r/w	WTV	r
DFONT	r	IDML	r	MPO	r/w	PSB	r/w	WV	r
DIVX	r	IIQ	r/w	MQV	r/w	PSD	r/w	X3F	r/w
DJVU	r	IND	r/w	MRC	r	PSP	r	XCF	r
DLL	r	INSP	r/w	MRW	r/w	QTIF	r/w	XLS	r
DNG	r/w	INSV	r	MXF	r	R3D	r	XLSX	r
DOC	r	INX	r	NEF	r/w	RA	r	XMP	r/w/c
DOCX	r	ISO	r	NKSC	r/w	RAF	r/w	ZIP	r

Meta Information

EXIF	r/w/c	CIFF	r/w	Ricoh RMETA	r
GPS	r/w/c	AFCP	r/w	Picture Info	r
IPTC	r/w/c	Kodak Meta	r/w	Adobe APP14	r
XMP	r/w/c	FotoStation	r/w	MPF	r
MakerNotes	r/w/c	PhotoMechanic	r/w	Stim	r
Photoshop IRB	r/w/c	JPEG 2000	r	DPX	r
ICC Profile	r/w/c	DICOM	r	APE	r
MIE	r/w/c	Flash	r	Vorbis	r
JFIF	r/w/c	FlashPix	r	SPIFF	r
Ducky APP12	r/w/c	QuickTime	r	DjVu	r
PDF	r/w/c	Matroska	r	M2TS	r
PNG	r/w/c	MXF	r	PE/COFF	r
Canon VRD	r/w/c	PrintIM	r	AVCHD	r
Nikon Capture	r/w/c	FLAC	r	ZIP	r
GeoTIFF	r/w/c	ID3	r	(and more)	

OPTIONS

Case is not significant for any command-line option (including tag and group names), except for single-character options when the corresponding upper-case option exists. Many single-character options have equivalent long-name versions (shown in brackets), and some options have inverses which are invoked with a leading double-dash. Unrecognized options are interpreted as tag names (for this reason, multiple single-character options may NOT be combined into one argument). Contrary to standard practice, options may appear after source file names on the exiftool command line.

Option Overview

Tag operations

-TAG or --TAG	Extract or exclude specified tag
-TAG[+-^]=[VALUE]	Write new value for tag
-TAG[+-]<=DATFILE	Write tag value from contents of file
-TAG[+-]<SRCTAG	Copy tag value (see -tagsFromFile)
-tagsFromFile SRCFILE	Copy tag values from file
-x TAG (-exclude)	Exclude specified tag

Input-output text formatting

-args (-argFormat)	Format metadata as exiftool arguments
-b (-binary)	Output metadata in binary format

```
  -c FMT      (-coordFormat)       Set format for GPS coordinates
  -charset [[TYPE=]CHARSET]        Specify encoding for special characters
  -csv[[+]=CSVFILE]                Export/import tags in CSV format
  -csvDelim STR                    Set delimiter for CSV file
  -d FMT      (-dateFormat)        Set format for date/time values
  -D          (-decimal)           Show tag ID numbers in decimal
  -E,-ex,-ec  (-escape(HTML|XML|C))Escape tag values for HTML, XML or C
  -f          (-forcePrint)        Force printing of all specified tags
  -g[NUM...]  (-groupHeadings)     Organize output by tag group
  -G[NUM...]  (-groupNames)        Print group name for each tag
  -h          (-htmlFormat)        Use HTML formatting for output
  -H          (-hex)               Show tag ID numbers in hexadecimal
  -htmlDump[OFFSET]                Generate HTML-format binary dump
  -j[[+]=JSONFILE] (-json)         Export/import tags in JSON format
  -l          (-long)              Use long 2-line output format
  -L          (-latin)             Use Windows Latin1 encoding
  -lang [LANG]                     Set current language
  -listItem INDEX                  Extract specific item from a list
  -n          (--printConv)        No print conversion
  -p FMTFILE  (-printFormat)       Print output in specified format
  -php                             Export tags as a PHP Array
  -s[NUM]     (-short)             Short output format
  -S          (-veryShort)         Very short output format
  -sep STR    (-separator)         Set separator string for list items
  -sort                            Sort output alphabetically
  -struct                          Enable output of structured information
  -t          (-tab)               Output in tab-delimited list format
  -T          (-table)             Output in tabular format
  -v[NUM]     (-verbose)           Print verbose messages
  -w[+|!] EXT (-textOut)           Write (or overwrite!) output text files
  -W[+|!] FMT (-tagOut)            Write output text file for each tag
  -Wext EXT   (-tagOutExt)         Write only specified file types with -W
  -X          (-xmlFormat)         Use RDF/XML output format

Processing control

  -a          (-duplicates)        Allow duplicate tags to be extracted
  -e          (--composite)        Do not generate composite tags
  -ee[NUM]    (-extractEmbedded)   Extract information from embedded files
  -ext[+] EXT (-extension)         Process files with specified extension
  -F[OFFSET]  (-fixBase)           Fix the base for maker notes offsets
  -fast[NUM]                       Increase speed when extracting metadata
  -fileOrder[NUM] [-]TAG           Set file processing order
  -i DIR      (-ignore)            Ignore specified directory name
  -if[NUM] EXPR                    Conditionally process files
  -m          (-ignoreMinorErrors) Ignore minor errors and warnings
  -o OUTFILE  (-out)               Set output file or directory name
  -overwrite_original              Overwrite original by renaming tmp file
  -overwrite_original_in_place     Overwrite original by copying tmp file
  -P          (-preserve)          Preserve file modification date/time
  -password PASSWD                 Password for processing protected files
  -progress[NUM][:[TITLE]]         Show file progress count
  -q          (-quiet)             Quiet processing
  -r[.]       (-recurse)           Recursively process subdirectories
  -scanForXMP                      Brute force XMP scan
  -u          (-unknown)           Extract unknown tags
  -U          (-unknown2)          Extract unknown binary tags too
  -wm MODE    (-writeMode)         Set mode for writing/creating tags
  -z          (-zip)               Read/write compressed information

Other options

  -@ ARGFILE                       Read command-line arguments from file
  -k          (-pause)             Pause before terminating
  -list[w|f|wf|g[NUM]|d|x]         List various exiftool capabilities
  -ver                             Print exiftool version number
  --                               End of options

Special features
```

```
  -geotag TRKFILE                 Geotag images from specified GPS log
  -globalTimeShift SHIFT          Shift all formatted date/time values
  -use MODULE                     Add features from plug-in module

Utilities

  -delete_original[!]             Delete "_original" backups
  -restore_original               Restore from "_original" backups

Advanced options

  -api OPT[[^]=[VAL]]             Set ExifTool API option
  -common_args                    Define common arguments
  -config CFGFILE                 Specify configuration file name
  -echo[NUM] TEXT                 Echo text to stdout or stderr
  -efile[NUM][!] ERRFILE          Save names of files with errors
  -execute[NUM]                   Execute multiple commands on one line
  -fileNUM ALTFILE                Load tags from alternate file
  -list_dir                       List directories, not their contents
  -srcfile FMT                    Process a different source file
  -stay_open FLAG                 Keep reading -@ argfile even after EOF
  -userParam PARAM[[^]=[VAL]]     Set user parameter (API UserParam opt)
```

Option Details

Tag operations

-TAG Extract information for the specified tag (eg. "-CreateDate"). Multiple tags may be specified in a single command. A tag name is the handle by which a piece of information is referenced. See Image::ExifTool::TagNames for documentation on available tag names. A tag name may include leading group names separated by colons (eg. "-EXIF:CreateDate", or "-Doc1:XMP:Creator"), and each group name may be prefixed by a digit to specify family number (eg. "-1IPTC:City"). (Note that the API SavePath and SaveFormat options must be used for the family 5 and 6 groups respectively to be available.) Use the **-listg** option to list available group names by family.

A special tag name of "All" may be used to indicate all meta information (ie. **-All**). This is particularly useful when a group name is specified to extract all information in a group (but beware that unless the **-a** option is also used, some tags in the group may be suppressed by same-named tags in other groups). The wildcard characters "?" and "*" may be used in a tag name to match any single character and zero or more characters respectively. These may not be used in a group name, with the exception that a group name of "*" (or "All") may be used to extract all instances of a tag (as if **-a** was used). Note that arguments containing wildcards must be quoted on the command line of most systems to prevent shell globbing.

A "#" may be appended to the tag name to disable the print conversion on a per-tag basis (see the **-n** option). This may also be used when writing or copying tags.

If no tags are specified, all available information is extracted (as if "-All" had been specified).

Note: Descriptions, not tag names, are shown by default when extracting information. Use the **-s** option to see the tag names instead.

--TAG

Exclude specified tag from extracted information. Same as the **-x** option. Group names and wildcards are permitted as described above for **-TAG**. Once excluded from the output, a tag may not be re-included by a subsequent option. May also be used following a **-tagsFromFile** option to exclude tags from being copied (when redirecting to another tag, it is the source tag that should be excluded), or to exclude groups from being deleted when deleting

all information (eg. "-all= --exif:all" deletes all but EXIF information). But note that this will not exclude individual tags from a group delete (unless a family 2 group is specified, see note 4 below). Instead, individual tags may be recovered using the **-tagsFromFile** option (eg. "-all= -tagsfromfile @ -artist").

To speed processing when reading XMP, exclusions in XMP groups also bypass processing of the corresponding XMP property and any contained properties. For example, "--xmp-crs:all" may speed processing significantly in cases where a large number of XMP-crs tags exist. To use this feature to bypass processing of a specific XMP property, the property name must be used instead of the ExifTool tag name (eg. "--xmp-crs:dabs"). Also, "XMP-all" may be used to to indicate any XMP namespace (eg. "--xmp-all:dabs").

-TAG[+-^]=[VALUE]

Write a new value for the specified tag (eg. "-comment=wow"), or delete the tag if no VALUE is given (eg. "-comment="). "+=" and "-=" are used to add or remove existing entries from a list, or to shift date/time values (see Image::ExifTool::Shift.pl and note 6 below for more details). "+=" may also be used to increment numerical values (or decrement if VALUE is negative), and "-=" may be used to conditionally delete or replace a tag (see "WRITING EXAMPLES" for examples). "^=" is used to write an empty string instead of deleting the tag when no VALUE is given, but otherwise it is equivalent to "=", but note that the caret must be quoted on the Windows command line.

TAG may contain one or more leading family 0, 1, 2 or 7 group names, prefixed by optional family numbers, and separated colons. If no group name is specified, the tag is created in the preferred group, and updated in any other location where a same-named tag already exists. The preferred group in JPEG and TIFF-format images is the first group in the following list where TAG is valid: 1) EXIF, 2) IPTC, 3) XMP.

The wildcards "*" and "?" may be used in tag names to assign the same value to multiple tags. When specified with wildcards, "Unsafe" tags are not written. A tag name of "All" is equivalent to "*" (except that it doesn't require quoting, while arguments with wildcards do on systems with shell globbing), and is often used when deleting all metadata (ie. "-All=") or an entire group (eg. "-XMP-dc:All=", see note 4 below). Note that not all groups are deletable, and that the JPEG APP14 "Adobe" group is not removed by default with "-All=" because it may affect the appearance of the image. However, color space information is removed, so the colors may be affected (but this may be avoided by copying back the tags defined by the ColorSpaceTags shortcut). Use the **-listd** option for a complete list of deletable groups, and see note 5 below regarding the "APP" groups. Also, within an image some groups may be contained within others, and these groups are removed if the containing group is deleted:

JPEG Image:
- Deleting EXIF or IFD0 also deletes ExifIFD, GlobParamIFD, GPS, IFD1, InteropIFD, MakerNotes, PrintIM and SubIFD.
- Deleting ExifIFD also deletes InteropIFD and MakerNotes.
- Deleting Photoshop also deletes IPTC.

TIFF Image:
- Deleting EXIF only removes ExifIFD which also deletes InteropIFD and MakerNotes.

MOV/MP4 Video:
- Deleting ItemList also deletes Keys tags.

Notes:

1) Many tag values may be assigned in a single command. If two assignments affect the same tag, the latter takes precedence

(except for list-type tags, for which both values are written).

2) In general, MakerNotes tags are considered "Permanent", and may be edited but not created or deleted individually. This avoids many potential problems, including the inevitable compatibility problems with OEM software which may be very inflexible about the information it expects to find in the maker notes.

3) Changes to PDF files by ExifTool are reversible (by deleting the update with "-PDF-update:all=") because the original information is never actually deleted from the file. So ExifTool alone may not be used to securely edit metadata in PDF files.

4) Specifying "-GROUP:all=" deletes the entire group as a block only if a single family 0 or 1 group is specified. Otherwise all deletable tags in the specified group(s) are removed individually, and in this case is it possible to exclude individual tags from a mass delete. For example, "-time:all --Exif:Time:All" removes all deletable Time tags except those in the EXIF. This difference also applies if family 2 is specified when deleting all groups. For example, "-2all:all=" deletes tags individually, while "-all:all=" deletes entire blocks.

5) The "APP" group names ("APP0" through "APP15") are used to delete JPEG application segments which are not associated with another deletable group. For example, specifying "-APP14:All=" will NOT delete the APP14 "Adobe" segment because this is accomplished with "-Adobe:All".

6) When shifting a value, the shift is applied to the original value of the tag, overriding any other values previously assigned to the tag on the same command line. To shift a date/time value and copy it to another tag in the same operation, use the **-globalTimeShift** option.

Special feature: Integer values may be specified in hexadecimal with a leading "0x", and simple rational values may be specified as fractions.

-TAG<=DATFILE or -TAG<=FMT
: Set the value of a tag from the contents of file DATFILE. The file name may also be given by a FMT string where %d, %f and %e represent the directory, file name and extension of the original FILE (see the **-w** option for more details). Note that quotes are required around this argument to prevent shell redirection since it contains a "<" symbol. If DATFILE/FMT is not provided, the effect is the same as "-TAG=", and the tag is simply deleted. "+<=" or "-<=" may also be used to add or delete specific list entries, or to shift date/time values.

-tagsFromFile SRCFILE or FMT
: Copy tag values from SRCFILE to FILE. Tag names on the command line after this option specify the tags to be copied, or excluded from the copy. Wildcards are permitted in these tag names. If no tags are specified, then all possible tags (see note 1 below) from the source file are copied to same-named tags in the preferred location of the output file (the same as specifying "-all"). More than one **-tagsFromFile** option may be used to copy tags from multiple files.

  By default, this option will update any existing and writable same-named tags in the output FILE, but will create new tags only in their preferred groups. This allows some information to be automatically transferred to the appropriate group when copying between images of different formats. However, if a group name is specified for a tag then the information is written only to this group (unless redirected to another group, see below). If "All" is used as a group name, then the specified tag(s) are written to the same family 1 group they had in the source file (ie. the same specific location, like ExifIFD or XMP-dc). For example, the

common operation of copying all writable tags to the same specific locations in the output FILE is achieved by adding "-all:all". A different family may be specified by adding a leading family number to the group name (eg. "-0all:all" preserves the same general location, like EXIF or XMP).

SRCFILE may be the same as FILE to move information around within a single file. In this case, "@" may be used to represent the source file (ie. "-tagsFromFile @"), permitting this feature to be used for batch processing multiple files. Specified tags are then copied from each file in turn as it is rewritten. For advanced batch use, the source file name may also be specified using a FMT string in which %d, %f and %e represent the directory, file name and extension of FILE. (eg. the current FILE would be represented by "%d%f.%e", with the same effect as "@"). See the **-w** option for FMT string examples.

A powerful redirection feature allows a destination tag to be specified for each copied tag. With this feature, information may be written to a tag with a different name or group. This is done using "'-DSTTAG<SRCTAG'" or "'-SRCTAG>DSTTAG'" on the command line after **-tagsFromFile**, and causes the value of SRCTAG to be copied from SRCFILE and written to DSTTAG in FILE. Has no effect unless SRCTAG exists in SRCFILE. Note that this argument must be quoted to prevent shell redirection, and there is no "=" sign as when assigning new values. Source and/or destination tags may be prefixed by a group name and/or suffixed by "#". Wildcards are allowed in both the source and destination tag names. A destination group and/or tag name of "All" or "*" writes to the same family 1 group and/or tag name as the source (but the family may be specified by adding a leading number to the group name, eg. "0All" writes to the same family 0 group as the source). If no destination group is specified, the information is written to the preferred group. Whitespace around the ">" or "<" is ignored. As a convenience, "-tagsFromFile @" is assumed for any redirected tags which are specified without a prior **-tagsFromFile** option. Copied tags may also be added or deleted from a list with arguments of the form "'-SRCTAG+<DSTTAG'" or "'-SRCTAG-<DSTTAG'" (but see Note 5 below).

An extension of the redirection feature allows strings involving tag names to be used on the right hand side of the "<" symbol with the syntax "'-DSTTAG<STR'", where tag names in STR are prefixed with a "$" symbol. See the **-p** option and the "Advanced formatting feature" section for more details about this syntax. Strings starting with a "=" sign must insert a single space after the "<" to avoid confusion with the "<=" operator which sets the tag value from the contents of a file. A single space at the start of the string is removed if it exists, but all other whitespace in the string is preserved. See note 8 below about using the redirection feature with list-type stags, shortcuts or when using wildcards in tag names.

See "COPYING EXAMPLES" for examples using **-tagsFromFile**.

Notes:

1) Some tags (generally tags which may affect the appearance of the image) are considered "Unsafe" to write, and are only copied if specified explicitly (ie. no wildcards). See the tag name documentation for more details about "Unsafe" tags.

2) Be aware of the difference between excluding a tag from being copied (--TAG), and deleting a tag (-TAG=). Excluding a tag prevents it from being copied to the destination image, but deleting will remove a pre-existing tag from the image.

3) The maker note information is copied as a block, so it isn't affected like other information by subsequent tag assignments on the command line, and individual makernote tags may not be

excluded from a block copy. Also, since the PreviewImage referenced from the maker notes may be rather large, it is not copied, and must be transferred separately if desired.

4) The order of operations is to copy all specified tags at the point of the **-tagsFromFile** option in the command line. Any tag assignment to the right of the **-tagsFromFile** option is made after all tags are copied. For example, new tag values are set in the order One, Two, Three then Four with this command:

```
exiftool -One=1 -tagsFromFile s.jpg -Two -Four=4 -Three d.jpg
```

This is significant in the case where an overlap exists between the copied and assigned tags because later operations may override earlier ones.

5) The normal behaviour of copied tags differs from that of assigned tags for list-type tags and conditional replacements because each copy operation on a tag overrides any previous operations. While this avoids duplicate list items when copying groups of tags from a file containing redundant information, it also prevents values of different tags from being copied into the same list when this is the intent. To accumulate values from different operations into the same list, add a "+" after the initial "-" of the argument. For example:

```
exiftool -tagsfromfile @ '-subject<make' '-+subject<model' ...
```

Similarly, "-+DSTTAG" must be used when conditionally replacing a tag to prevent overriding earlier conditions.

6) The **-a** option (allow duplicate tags) is always in effect when copying tags from SRCFILE, but the highest priority tag is always copied last so it takes precedence.

7) Structured tags are copied by default when copying tags. See the **-struct** option for details.

8) With the redirection feature, copying a tag directly (ie. "'-DSTTAG<SRCTAG'") is not the same as interpolating its value inside a string (ie. "'-DSTTAG<$SRCTAG'") for source tags which are list-type tags, shortcut tags, tag names containing wildcards, or UserParam variables. When copying directly, the values of each matching source tag are copied individually to the destination tag (as if they were separate assignments). However, when interpolated inside a string, list items and the values of shortcut tags are concatenated (with a separator set by the **-sep** option), and wildcards are not allowed. Also, UserParam variables are available only when interpolated in a string. Another difference is that a minor warning is generated if a tag doesn't exist when interpolating its value in a string (with "$"), but isn't when copying the tag directly.

Finally, the behaviour is different when a destination tag or group of "All" is used. When copying directly, a destination group and/or tag name of "All" writes to the same family 1 group and/or tag name as the source. But when interpolated in a string, the identity of the source tags are lost and the value is written to all possible groups/tags. For example, the string form must be used in the following command since the intent is to set the value of all existing date/time tags from "CreateDate":

```
exiftool '-time:all<$createdate' -wm w FILE
```

-x TAG (**-exclude**)
: Exclude the specified tag. There may be multiple -x options. This has the same effect as --TAG on the command line. See the --TAG documentation above for a complete description.

Input-output text formatting

Note that trailing spaces are removed from extracted values for most output text formats. The exceptions are -b, -csv, -j and -X.

-args (-argFormat)
Output information in the form of exiftool arguments, suitable for use with the -@ option when writing. May be combined with the -G option to include group names. This feature may be used to effectively copy tags between images, but allows the metadata to be altered by editing the intermediate file ("out.args" in this example):

```
exiftool -args -G1 --filename --directory src.jpg > out.args
exiftool -@ out.args -sep ', ' dst.jpg
```

Note: Be careful when copying information with this technique since it is easy to write tags which are normally considered "Unsafe". For instance, the FileName and Directory tags are excluded in the example above to avoid renaming and moving the destination file. Also note that the second command above will produce warning messages for any tags which are not writable.

As well, the -sep option should be used as in the second command above to maintain separate list items when writing metadata back to image files, and the -struct option may be used when extracting to preserve structured XMP information.

-b, --b (-binary, --binary)
Output requested metadata in binary format without tag names or descriptions (-b or -binary). This option is mainly used for extracting embedded images or other binary data, but it may also be useful for some text strings since control characters (such as newlines) are not replaced by '.' as they are in the default output. By default, list items are separated by a newline when extracted with the -b option, but this may be changed (see the -sep option for details). May be combined with -j, -php or -X to extract binary data in JSON, PHP or XML format, but note that "Unsafe" tags are not extracted as binary unless they are specified explicitly or the API RequestAll option is set to 3 or higher.

With a leading double dash (--b or --binary), tags which contain binary data are suppressed in the output when reading.

-c FMT (-coordFormat)
Set the print format for GPS coordinates. FMT uses the same syntax as a "printf" format string. The specifiers correspond to degrees, minutes and seconds in that order, but minutes and seconds are optional. For example, the following table gives the output for the same coordinate using various formats:

FMT	Output	
"%d deg %d' %.2f"\"	54 deg 59' 22.80"	(default for reading)
"%d %d %.8f"	54 59 22.80000000	(default for copying)
"%d deg %.4f min"	54 deg 59.3800 min	
"%.6f degrees"	54.989667 degrees	

Notes:

1) To avoid loss of precision, the default coordinate format is different when copying tags using the -tagsFromFile option.

2) If the hemisphere is known, a reference direction (N, S, E or W) is appended to each printed coordinate, but adding a "+" to the format specifier (eg. "%+.6f") prints a signed coordinate instead.

3) This print formatting may be disabled with the -n option to extract coordinates as signed decimal degrees.

-charset [[TYPE=]CHARSET]

If TYPE is "ExifTool" or not specified, this option sets the ExifTool character encoding for output tag values when reading and input values when writing, with a default of "UTF8". If no CHARSET is given, a list of available character sets is returned. Valid CHARSET values are:

CHARSET	Alias(es)	Description
UTF8	cp65001, UTF-8	UTF-8 characters (default)
Latin	cp1252, Latin1	Windows Latin1 (West European)
Latin2	cp1250	Windows Latin2 (Central European)
Cyrillic	cp1251, Russian	Windows Cyrillic
Greek	cp1253	Windows Greek
Turkish	cp1254	Windows Turkish
Hebrew	cp1255	Windows Hebrew
Arabic	cp1256	Windows Arabic
Baltic	cp1257	Windows Baltic
Vietnam	cp1258	Windows Vietnamese
Thai	cp874	Windows Thai
DOSLatinUS	cp437	DOS Latin US
DOSLatin1	cp850	DOS Latin1
DOSCyrillic	cp866	DOS Cyrillic
MacRoman	cp10000, Roman	Macintosh Roman
MacLatin2	cp10029	Macintosh Latin2 (Central Europe)
MacCyrillic	cp10007	Macintosh Cyrillic
MacGreek	cp10006	Macintosh Greek
MacTurkish	cp10081	Macintosh Turkish
MacRomanian	cp10010	Macintosh Romanian
MacIceland	cp10079	Macintosh Icelandic
MacCroatian	cp10082	Macintosh Croatian

TYPE may be "FileName" to specify the encoding of file names on the command line (ie. FILE arguments). In Windows, this triggers use of wide-character i/o routines, thus providing support for Unicode file names. See the "WINDOWS UNICODE FILE NAMES" section below for details.

Other values of TYPE listed below are used to specify the internal encoding of various meta information formats.

TYPE	Description	Default
EXIF	Internal encoding of EXIF "ASCII" strings	(none)
ID3	Internal encoding of ID3v1 information	Latin
IPTC	Internal IPTC encoding to assume when IPTC:CodedCharacterSet is not defined	Latin
Photoshop	Internal encoding of Photoshop IRB strings	Latin
QuickTime	Internal encoding of QuickTime strings	MacRoman
RIFF	Internal encoding of RIFF strings	0

See <https://exiftool.org/faq.html#Q10> for more information about coded character sets, and the Image::ExifTool Options for more details about the **-charset** settings.

-csv[[+]=CSVFILE]

Export information in CSV format, or import information if CSVFILE is specified. When importing, the CSV file must be in exactly the same format as the exported file. The first row of the CSVFILE must be the ExifTool tag names (with optional group names) for each column of the file, and values must be separated by commas. A special "SourceFile" column specifies the files associated with each row of information (and a SourceFile of "*" may be used to define default tags to be imported for all files which are combined with any tags specified for the specific SourceFile processed). The **-csvDelim** option may be used to change the input/output field delimiter if something other than a comma is required.

The following examples demonstrate basic use of the **-csv** option:

```
    # generate CSV file with common tags from all images in a directory
    exiftool -common -csv dir > out.csv

    # update metadata for all images in a directory from CSV file
    exiftool -csv=a.csv dir
```

When importing, empty values are ignored unless the **-f** option is used and the API MissingTagValue is set to an empty string (in which case the tag is deleted). Also, FileName and Directory columns are ignored if they exist (ie. ExifTool will not attempt to write these tags with a CSV import), but all other columns are imported. To force a tag to be deleted, use the **-f** option and set the value to "-" in the CSV file (or to the MissingTagValue if this API option was used). Multiple databases may be imported in a single command.

When exporting a CSV file, the **-g** or **-G** option adds group names to the tag headings. If the **-a** option is used to allow duplicate tag names, the duplicate tags are only included in the CSV output if the column headings are unique. Adding the **-G4** option ensures a unique column heading for each tag. The **-b** option may be added to output binary data, encoded in base64 if necessary (indicated by ASCII "base64:" as the first 7 bytes of the value). Values may also be encoded in base64 if the **-charset** option is used and the value contains invalid characters.

When exporting specific tags, the CSV columns are arranged in the same order as the specified tags provided the column headings exactly match the specified tag names, otherwise the columns are sorted in alphabetical order.

When importing from a CSV file, only files specified on the command line are processed. Any extra entries in the CSV file are ignored.

List-type tags are stored as simple strings in a CSV file, but the **-sep** option may be used to split them back into separate items when importing.

Special feature: **-csv**+=CSVFILE may be used to add items to existing lists. This affects only list-type tags. Also applies to the **-j** option.

Note that this option is fundamentally different than all other output format options because it requires information from all input files to be buffered in memory before the output is written. This may result in excessive memory usage when processing a very large number of files with a single command. Also, it makes this option incompatible with the **-w** and **-W** options. When processing a large number of files, it is recommended to either use the JSON (**-j**) or XML (**-X**) output format, or use **-p** to generate a fixed-column CSV file instead of using the **-csv** option.

-csvDelim STR

Set the delimiter for separating CSV entries for CSV file input/output via the **-csv** option. STR may contain "\t", "\n", "\r" and "\\" to represent TAB, LF, CR and '\' respectively. A double quote is not allowed in the delimiter. Default is ','.

-d FMT (**-dateFormat**)

Set the format for date/time tag values. The FMT string may contain formatting codes beginning with a percent character ("%") to represent the various components of a date/time value. The specifics of the FMT syntax are system dependent -- consult the "strftime" man page on your system for details. The default format is equivalent to "%Y:%m:%d %H:%M:%S". This option has no effect on date-only or time-only tags and ignores timezone information if present. ExifTool adds a %f format code to represent fractional seconds, and supports an optional width to

specify the number of digits after the decimal point (eg. %3f would give something like .437), and a minus sign to drop the decimal point (eg. "%-3f" would give 437). Only one **-d** option may be used per command. Requires POSIX::strptime or Time::Piece for the inversion conversion when writing.

-D (**-decimal**)
Show tag ID number in decimal when extracting information.

-E, **-ex**, **-ec** (**-escapeHTML**, **-escapeXML**, **-escapeC**)
Escape characters in output tag values for HTML (**-E**), XML (**-ex**) or C (**-ec**). For HTML, all characters with Unicode code points above U+007F are escaped as well as the following 5 characters: & (&) ' (') " (") > (>) and < (<). For XML, only these 5 characters are escaped. The **-E** option is implied with **-h**, and **-ex** is implied with **-X**. For C, all control characters and the backslash are escaped. The inverse conversion is applied when writing tags.

-f (**-forcePrint**)
Force printing of tags even if their values are not found. This option only applies when specific tags are requested on the command line (ie. not with wildcards or by "-all"). With this option, a dash ("-") is printed for the value of any missing tag, but the dash may be changed via the API MissingTagValue option. May also be used to add a 'flags' attribute to the **-listx** output, or to allow tags to be deleted when writing with the **-csv**=CSVFILE feature.

-g[NUM][:NUM...] (**-groupHeadings**)
Organize output by tag group. NUM specifies a group family number, and may be 0 (general location), 1 (specific location), 2 (category), 3 (document number), 4 (instance number), 5 (metadata path), 6 (EXIF/TIFF format), 7 (tag ID) or 8 (file number). **-g0** is assumed if a family number is not specified. May be combined with other options to add group names to the output. Multiple families may be specified by separating them with colons. By default the resulting group name is simplified by removing any leading "Main:" and collapsing adjacent identical group names, but this can be avoided by placing a colon before the first family number (eg. **-g:3:1**). Use the **-listg** option to list group names for a specified family. The API SavePath and SaveFormat options are automatically enabled if the respective family 5 or 6 group names are requested. See the API GetGroup documentation for more information.

-G[NUM][:NUM...] (**-groupNames**)
Same as **-g** but print group name for each tag. **-G0** is assumed if NUM is not specified. May be combined with a number of other options to add group names to the output. Note that NUM may be added wherever **-G** is mentioned in the documentation. See the **-g** option above for details.

-h (**-htmlFormat**)
Use HTML table formatting for output. Implies the **-E** option. The formatting options **-D**, **-H**, **-g**, **-G**, **-l** and **-s** may be used in combination with **-h** to influence the HTML format.

-H (**-hex**)
Show tag ID number in hexadecimal when extracting information.

-htmlDump[OFFSET]
Generate a dynamic web page containing a hex dump of the EXIF information. This can be a very powerful tool for low-level analysis of EXIF information. The **-htmlDump** option is also invoked if the **-v** and **-h** options are used together. The verbose level controls the maximum length of the blocks dumped. An OFFSET may be given to specify the base for displayed offsets. If not provided, the EXIF/TIFF base offset is used. Use **-htmlDump0** for absolute offsets. Currently only EXIF/TIFF and JPEG information

is dumped, but the -u option can be used to give a raw hex dump of other file formats.

-j[[+]=JSONFILE] (-json)
Use JSON (JavaScript Object Notation) formatting for console output, or import JSON file if JSONFILE is specified. This option may be combined with -g to organize the output into objects by group, or -G to add group names to each tag. List-type tags with multiple items are output as JSON arrays unless -sep is used. By default XMP structures are flattened into individual tags in the JSON output, but the original structure may be preserved with the -struct option (this also causes all list-type XMP tags to be output as JSON arrays, otherwise single-item lists would be output as simple strings). The -a option is implied when -json is used, but entries with identical JSON names are suppressed in the output. (-G4 may be used to ensure that all tags have unique JSON names.) Adding the -D or -H option changes tag values to JSON objects with "val" and "id" fields, and adding -l adds a "desc" field, and a "num" field if the numerical value is different from the converted "val". The -b option may be added to output binary data, encoded in base64 if necessary (indicated by ASCII "base64:" as the first 7 bytes of the value), and -t may be added to include tag table information (see -t for details). The JSON output is UTF-8 regardless of any -L or -charset option setting, but the UTF-8 validation is disabled if a character set other than UTF-8 is specified. Note that ExifTool quotes JSON values only if they don't look like numbers (regardless of the original storage format or the relevant metadata specification).

If JSONFILE is specified, the file is imported and the tag definitions from the file are used to set tag values on a per-file basis. The special "SourceFile" entry in each JSON object associates the information with a specific target file. An object with a missing SourceFile or a SourceFile of "*" defines default tags for all target files which are combined with any tags specified for the specific SourceFile processed. The imported JSON file must have the same format as the exported JSON files with the exception that options exporting JSON objects instead of simple values are not compatible with the import file format (ie. export with -D, -H, -l, or -T is not compatable, and use -G instead of -g). Additionally, tag names in the input JSON file may be suffixed with a "#" to disable print conversion.

Unlike CSV import, empty values are not ignored, and will cause an empty value to be written if supported by the specific metadata type. Tags are deleted by using the -f option and setting the tag value to "-" (or to the MissingTagValue setting if this API option was used). Importing with -j+=JSONFILE causes new values to be added to existing lists.

-l (-long)
Use long 2-line Canon-style output format. Adds a description and unconverted value (if it is different from the converted value) to the XML, JSON or PHP output when -X, -j or -php is used. May also be combined with -listf, -listr or -listwf to add descriptions of the file types.

-L (-latin)
Use Windows Latin1 encoding (cp1252) for output tag values instead of the default UTF-8. When writing, -L specifies that input text values are Latin1 instead of UTF-8. Equivalent to "-charset latin".

-lang [LANG]
Set current language for tag descriptions and converted values. LANG is "de", "fr", "ja", etc. Use -lang with no other arguments to get a list of available languages. The default language is "en" if -lang is not specified. Note that tag/group names are always English, independent of the -lang setting, and translation of warning/error messages has not yet been implemented. May also

be combined with -listx to output descriptions in one language only.

By default, ExifTool uses UTF-8 encoding for special characters, but the -L or -charset option may be used to invoke other encodings. Note that ExifTool uses Unicode::LineBreak if available to help preserve the column alignment of the plain text output for languages with a variable-width character set.

Currently, the language support is not complete, but users are welcome to help improve this by submitting their own translations. To submit a translation, follow these steps (you must have Perl installed for this):

1. Download and unpack the latest Image-ExifTool full distribution.

2. 'cd' into the Image-ExifTool directory.

3. Run this command to make an XML file of the desired tags (eg. EXIF):

```
./exiftool -listx -exif:all > out.xml
```

4. Copy this text into a file called 'import.pl' in the exiftool directory:

```
push @INC, 'lib';
require Image::ExifTool::TagInfoXML;
my $file = shift or die "Expected XML file name\n";
$Image::ExifTool::TagInfoXML::makeMissing = shift;
Image::ExifTool::TagInfoXML::BuildLangModules($file,8);
```

5. Run the 'import.pl' script to Import the XML file, generating the 'MISSING' entries for your language (eg. Russian):

```
perl import.pl out.xml ru
```

6. Edit the generated language module lib/Image/ExifTool/Lang/ru.pm, and search and replace all 'MISSING' strings in the file with your translations.

7. Email the module ('ru.pm' in this example) to philharvey66 at gmail.com

8. Thank you!!

-listItem INDEX
: For list-type tags, this causes only the item with the specified index to be extracted. INDEX is 0 for the first item in the list. Negative indices may also be used to reference items from the end of the list. Has no effect on single-valued tags. Also applies to tag values when copying from a tag, and in -if conditions.

-n (--printConv)
: Disable print conversion for all tags. By default, extracted values are converted to a more human-readable format, but the -n option disables this conversion, revealing the machine-readable values. For example:

```
> exiftool -Orientation -S a.jpg
Orientation: Rotate 90 CW
> exiftool -Orientation -S -n a.jpg
Orientation: 6
```

The print conversion may also be disabled on a per-tag basis by suffixing the tag name with a "#" character:

```
> exiftool -Orientation# -Orientation -S a.jpg
Orientation: 6
```

```
    Orientation: Rotate 90 CW
```

These techniques may also be used to disable the inverse print conversion when writing. For example, the following commands all have the same effect:

```
    > exiftool -Orientation='Rotate 90 CW' a.jpg
    > exiftool -Orientation=6 -n a.jpg
    > exiftool -Orientation#=6 a.jpg
```

-p FMTFILE or STR (-printFormat)

Print output in the format specified by the given file or string. The argument is interpreted as a string unless a file of that name exists, in which case the string is loaded from the contents of the file. Tag names in the format file or string begin with a "$" symbol and may contain leading group names and/or a trailing "#" (to disable print conversion). Case is not significant. Braces "{}" may be used around the tag name to separate it from subsequent text (and must be used if subsequent text begins with an alphanumeric character, hyphen, underline, colon or number sign). Use $$ to represent a "$" symbol, and $/ for a newline.

Multiple -p options may be used, each contributing a line (or more) of text to the output. Lines beginning with "#[HEAD]" and "#[TAIL]" are output before the first processed file and after the last processed file respectively. Lines beginning with "#[SECT]" and "#[ENDS]" are output before and after each section of files. A section is defined as a group of consecutive files with the same section header (eg. files are grouped by directory if "#[SECT]" contains $directory). Lines beginning with "#[BODY]" and lines not beginning with "#" are output for each processed file. Lines beginning with "#[IF]" are not output, but all BODY lines are skipped if any tag on an IF line doesn't exist. Other lines beginning with "#" are ignored. (To output a line beginning with "#", use "#[BODY]#".) For example, this format file:

```
    # this is a comment line
    #[HEAD]-- Generated by ExifTool $exifToolVersion --
    File: $FileName - $DateTimeOriginal
    (f/$Aperture, ${ShutterSpeed}s, ISO $EXIF:ISO)
    #[TAIL]-- end --
```

with this command:

```
    exiftool -p test.fmt a.jpg b.jpg
```

produces output like this:

```
    -- Generated by ExifTool 12.59 --
    File: a.jpg - 2003:10:31 15:44:19
    (f/5.6, 1/60s, ISO 100)
    File: b.jpg - 2006:05:23 11:57:38
    (f/8.0, 1/13s, ISO 100)
    -- end --
```

The values of List-type tags with multiple items and Shortcut tags representing multiple tags are joined according the -sep option setting when interpolated in the string.

When -ee (-extractEmbedded) is combined with -p, embedded documents are effectively processed as separate input files.

If a specified tag does not exist, a minor warning is issued and the line with the missing tag is not printed. However, the -f option may be used to set the value of missing tags to '-' (but this may be configured via the API MissingTagValue option), or the -m option may be used to ignore minor warnings and leave the missing values empty. Alternatively, -q -q may be used to simply suppress the warning messages.

The "Advanced formatting feature" may be used to modify the values of individual tags within the -p option string.

Note that the API RequestTags option is automatically set for all tags used in the FMTFILE or STR. This allows all other tags to be ignored using -API IgnoreTags=all, resulting in reduced memory usage and increased speed.

-php Format output as a PHP Array. The -g, -G, -D, -H, -l, -sep and -struct options combine with -php, and duplicate tags are handled in the same way as with the -json option. As well, the -b option may be added to output binary data, and -t may be added to include tag table information (see -t for details). Here is a simple example showing how this could be used in a PHP script:

```
<?php
eval('$array=' . `exiftool -php -q image.jpg`);
print_r($array);
?>
```

-s[NUM] (-short)
Short output format. Prints tag names instead of descriptions. Add NUM or up to 3 -s options for even shorter formats:

```
-s1 or -s        - print tag names instead of descriptions
-s2 or -s -s     - no extra spaces to column-align values
-s3 or -s -s -s  - print values only (no tag names)
```

Also effective when combined with -t, -h, -X or -listx options.

-S (-veryShort)
Very short format. The same as -s2 or two -s options. Tag names are printed instead of descriptions, and no extra spaces are added to column-align values.

-sep STR (-separator)
Specify separator string for items in list-type tags. When reading, the default is to join list items with ", ". When writing, this option causes values assigned to list-type tags to be split into individual items at each substring matching STR (otherwise they are not split by default). Space characters in STR match zero or more whitespace characters in the value.

Note that an empty separator ("") is allowed, and will join items with no separator when reading, or split the value into individual characters when writing.

For pure binary output (-b used without -j, -php or -X), the first -sep option specifies a list-item separator, and a second -sep option specifies a terminator for the end of the list (or after each value if not a list). In these strings, "\n", "\r" and "\t" may be used to represent a newline, carriage return and tab respectively. By default, binary list items are separated by a newline, and no terminator is added.

-sort, --sort
Sort output by tag description, or by tag name if the -s option is used. When sorting by description, the sort order will depend on the -lang option setting. Without the -sort option, tags appear in the order they were specified on the command line, or if not specified, the order they were extracted from the file. By default, tags are organized by groups when combined with the -g or -G option, but this grouping may be disabled with --sort.

-struct, --struct
Output structured XMP information instead of flattening to individual tags. This option works well when combined with the XML (-X) and JSON (-j) output formats. For other output formats, XMP structures and lists are serialized into the same format as when writing structured information (see

<https://exiftool.org/struct.html> for details). When copying, structured tags are copied by default unless **--struct** is used to disable this feature (although flattened tags may still be copied by specifying them individually unless **-struct** is used). These options have no effect when assigning new values since both flattened and structured tags may always be used when writing.

-t (**-tab**)
Output a tab-delimited list of description/values (useful for database import). May be combined with **-s** to print tag names instead of descriptions, or **-S** to print tag values only, tab-delimited on a single line. The **-t** option may be combined with **-j**, **-php** or **-X** to add tag table information ("table", tag "id", and "index" for cases where multiple conditional tags exist with the same ID).

-T (**-table**)
Output tag values in table form. Equivalent to **-t -S -q -f**.

-v[NUM] (**-verbose**)
Print verbose messages. NUM specifies the level of verbosity in the range 0-5, with higher numbers being more verbose. If NUM is not given, then each **-v** option increases the level of verbosity by 1. With any level greater than 0, most other options are ignored and normal console output is suppressed unless specific tags are extracted. Using **-v0** causes the console output buffer to be flushed after each line (which may be useful to avoid delays when piping exiftool output), and prints the name of each processed file when writing and the new file name when renaming, moving or copying. Verbose levels above **-v0** do not flush after each line. Also see the **-progress** option.

-w[+|!] EXT or FMT (**-textOut**)
Write console output to files with names ending in EXT, one for each source file. The output file name is obtained by replacing the source file extension (including the '.') with the specified extension (and a '.' is added to the start of EXT if it doesn't already contain one). Alternatively, a FMT string may be used to give more control over the output file name and directory. In the format string, %d, %f and %e represent the directory, filename and extension of the source file, and %c represents a copy number which is automatically incremented if the file already exists. %d includes the trailing '/' if necessary, but %e does not include the leading '.'. For example:

```
-w %d%f.txt       # same effect as "-w txt"
-w dir/%f_%e.out  # write files to "dir" as "FILE_EXT.out"
-w dir2/%d%f.txt  # write to "dir2", keeping dir structure
-w a%c.txt        # write to "a.txt" or "a1.txt" or "a2.txt"...
```

Existing files will not be changed unless an exclamation point is added to the option name (ie. **-w!** or **-textOut!**) to overwrite the file, or a plus sign (ie. **-w+** or **-textOut+**) to append to the existing file. Both may be used (ie. **-w+!** or **-textOut+!**) to overwrite output files that didn't exist before the command was run, and append the output from multiple source files. For example, to write one output file for all source files in each directory:

```
exiftool -filename -createdate -T -w+! %d/out.txt -r DIR
```

Capitalized format codes %D, %F, %E and %C provide slightly different alternatives to the lower case versions. %D does not include the trailing '/', %F is the full filename including extension, %E includes the leading '.', and %C increments the count for each processed file (see below).

Notes:

1) In a Windows BAT file the "%" character is represented by "%%",

so an argument like "%d%f.txt" is written as "%%d%%f.txt".

2) If the argument for **-w** does not contain a valid format code (eg. %f), then it is interpreted as a file extension, but there are three different ways to create a single output file from multiple source files:

```
# 1. Shell redirection
exiftool FILE1 FILE2 ... > out.txt

# 2. With the -w option and a zero-width format code
exiftool -w+! %0fout.txt FILE1 FILE2 ...

# 3. With the -W option (see the -W option below)
exiftool -W+! out.txt FILE1 FILE2 ...
```

Advanced features:

A substring of the original file name, directory or extension may be taken by specifying a field width immediately following the '%' character. If the width is negative, the substring is taken from the end. The substring position (characters to ignore at the start or end of the string) may be given by a second optional value after a decimal point. For example:

Input File Name	Format Specifier	Output File Name
Picture-123.jpg	%7f.txt	Picture.txt
Picture-123.jpg	%-.4f.out	Picture.out
Picture-123.jpg	%7f.%-3f	Picture.123
Picture-123a.jpg	Meta%-3.1f.txt	Meta123.txt

(Note that special characters may have a width of greater than one.)

For %d and %D, the field width/position specifiers may be applied to the directory levels instead of substring position by using a colon instead of a decimal point in the format specifier. For example:

Source Dir	Format	Result	Notes
pics/2012/02	%2:d	pics/2012/	take top 2 levels
pics/2012/02	%-:1d	pics/2012/	up one directory level
pics/2012/02	%:1d	2012/02/	ignore top level
pics/2012/02	%1:1d	2012/	take 1 level after top
pics/2012/02	%-1:D	02	bottom level folder name
/Users/phil	%:2d	phil/	ignore top 2 levels

(Note that the root directory counts as one level when an absolute path is used as in the last example above.)

For %c, these modifiers have a different effects. If a field width is given, the copy number is padded with zeros to the specified width. A leading '-' adds a dash before the copy number, and a '+' adds an underline. By default, the copy number is omitted from the first file of a given name, but this can be changed by adding a decimal point to the modifier. For example:

```
-w A%-cZ.txt      # AZ.txt, A-1Z.txt, A-2Z.txt ...
-w B%5c.txt       # B.txt, B00001.txt, B00002.txt ...
-w C%.c.txt       # C0.txt, C1.txt, C2.txt ...
-w D%-.c.txt      # D-0.txt, D-1.txt, D-2.txt ...
-w E%-.4c.txt     # E-0000.txt, E-0001.txt, E-0002.txt ...
-w F%-.4nc.txt    # F-0001.txt, F-0002.txt, F-0003.txt ...
-w G%+c.txt       # G.txt, G_1.txt G_2.txt ...
-w H%-lc.txt      # H.txt, H-b.txt, H-c.txt ...
-w I.%.3uc.txt    # I.AAA.txt, I.AAB.txt, I.AAC.txt ...
```

A special feature allows the copy number to be incremented for

each processed file by using %C (upper case) instead of %c. This allows a sequential number to be added to output file names, even if the names are different. For %C, a copy number of zero is not omitted as it is with %c. A leading '-' causes the number to be reset at the start of each new directory, and '+' has no effect. The number before the decimal place gives the starting index, the number after the decimal place gives the field width. The following examples show the output filenames when used with the command "exiftool rose.jpg star.jpg jet.jpg ...":

```
-w %C%f.txt       # 0rose.txt, 1star.txt, 2jet.txt
-w %f-%10C.txt    # rose-10.txt, star-11.txt, jet-12.txt
-w %.3C-%f.txt    # 000-rose.txt, 001-star.txt, 002-jet.txt
-w %57.4C%f.txt   # 0057rose.txt, 0058star.txt, 0059jet.txt
```

All format codes may be modified by 'l' or 'u' to specify lower or upper case respectively (ie. %le for a lower case file extension). When used to modify %c or %C, the numbers are changed to an alphabetical base (see example H above). Also, %c and %C may be modified by 'n' to count using natural numbers starting from 1, instead of 0 (see example F above).

This same FMT syntax is used with the **-o** and **-tagsFromFile** options, although %c and %C are only valid for output file names.

-W[+|!] FMT (**-tagOut**)

This enhanced version of the **-w** option allows a separate output file to be created for each extracted tag. See the **-w** option documentation above for details of the basic functionality. Listed here are the differences between **-W** and **-w**:

1) With **-W**, a new output file is created for each extracted tag.

2) **-W** supports four additional format codes: %t, %g and %s represent the tag name, group name, and suggested extension for the output file (based on the format of the data), and %o represents the value of the OriginalRawFileName or OriginalFileName tag from the input file (including extension). The %g code may be followed by a single digit to specify the group family number (eg. %g1), otherwise family 0 is assumed. The substring width/position/case specifiers may be used with these format codes in exactly the same way as with %f and %e.

3) The argument for **-W** is interpreted as a file name if it contains no format codes. (For **-w**, this would be a file extension.) This change allows a simple file name to be specified, which, when combined with the append feature, provides a method to write metadata from multiple source files to a single output file without the need for shell redirection. For example, the following pairs of commands give the same result:

```
# overwriting existing text file
exiftool test.jpg > out.txt     # shell redirection
exiftool test.jpg -W+! out.txt  # equivalent -W option

# append to existing text file
exiftool test.jpg >> out.txt    # shell redirection
exiftool test.jpg -W+ out.txt   # equivalent -W option
```

4) Adding the **-v** option to **-W** sends a list of the tags and output file names to the console instead of giving a verbose dump of the entire file. (Unless appending all output to one file for each source file by using **-W+** with an output file FMT that does not contain %t, %g, %s or %o.)

5) Individual list items are stored in separate files when **-W** is combined with **-b**, but note that for separate files to be created %c or %C must be used in FMT to give the files unique names.

-Wext EXT, **--Wext** EXT (**-tagOutExt**)

This option is used to specify the type of output file(s) written by the -W option. An output file is written only if the suggested extension matches EXT. Multiple -Wext options may be used to write more than one type of file. Use --Wext to write all but the specified type(s).

-X (-xmlFormat)
Use ExifTool-specific RDF/XML formatting for console output. Implies the -a option, so duplicate tags are extracted. The formatting options -b, -D, -H, -l, -s, -sep, -struct and -t may be used in combination with -X to affect the output, but note that the tag ID (-D, -H and -t), binary data (-b) and structured output (-struct) options are not effective for the short output (-s). Another restriction of -s is that only one tag with a given group and name may appear in the output. Note that the tag ID options (-D, -H and -t) will produce non-standard RDF/XML unless the -l option is also used.

By default, -X outputs flattened tags, so -struct should be added if required to preserve XMP structures. List-type tags with multiple values are formatted as an RDF Bag, but they are combined into a single string when -s or -sep is used. Using -L changes the XML encoding from "UTF-8" to "windows-1252". Other -charset settings change the encoding only if there is a corresponding standard XML character set. The -b option causes binary data values to be written, encoded in base64 if necessary. The -t option adds tag table information to the output (see -t for details).

Note: This output is NOT the same as XMP because it uses dynamically-generated property names corresponding to the ExifTool tag names with ExifTool family 1 group names as namespaces, and not the standard XMP properties and namespaces. To write XMP instead, use the -o option with an XMP extension for the output file.

Processing control

-a, --a (-duplicates, --duplicates)
Allow (-a) or suppress (--a) duplicate tag names to be extracted. By default, duplicate tags are suppressed when reading unless the -ee or -X options are used or the Duplicates option is enabled in the configuration file. When writing, this option allows multiple Warning messages to be shown. Duplicate tags are always extracted when copying.

-e (--composite)
Extract existing tags only -- don't generate composite tags.

-ee[NUM] (-extractEmbedded)
Extract information from embedded documents in EPS files, embedded EPS information and JPEG and Jpeg2000 images in PDF files, embedded MPF images in JPEG and MPO files, streaming metadata in AVCHD videos, and the resource fork of Mac OS files. Implies the -a option. Use -g3 or -G3 to identify the originating document for extracted information. Embedded documents containing sub-documents are indicated with dashes in the family 3 group name. (eg. "Doc2-3" is the 3rd sub-document of the 2nd embedded document.) Note that this option may increase processing time substantially, especially for PDF files with many embedded images or videos with streaming metadata.

When used with -ee, the -p option is evaluated for each embedded document as if it were a separate input file. This allows, for example, generation of GPS track logs from timed metadata in videos. See <https://exiftool.org/geotag.html#Inverse> for examples.

Setting NUM to 2 causes the H264 video stream in MP4 videos to be parsed until the first Supplemental Enhancement Information (SEI)

message is decoded, or 3 to parse the entire H624 stream and decode all SEI information. For M2TS videos, a setting of 3 causes the entire file to be parsed in search of unlisted programs which may contain timed GPS.

-ext[+] EXT, --ext EXT (-extension)
Process only files with (-ext) or without (--ext) a specified extension. There may be multiple -ext and --ext options. A plus sign may be added (ie. -ext+) to add the specified extension to the normally processed files. EXT may begin with a leading '.', which is ignored. Case is not significant. "*" may be used to process files with any extension (or none at all), as in the last three examples:

```
exiftool -ext JPG DIR             # process only JPG files
exiftool --ext cr2 --ext dng DIR  # supported files but CR2/DNG
exiftool -ext+ txt DIR            # supported files plus TXT
exiftool -ext "*" DIR             # process all files
exiftool -ext "*" --ext xml DIR   # process all but XML files
exiftool -ext "*" --ext . DIR     # all but those with no ext
```

Using this option has two main advantages over specifying "*.EXT" on the command line: 1) It applies to files in subdirectories when combined with the -r option. 2) The -ext option is case-insensitive, which is useful when processing files on case-sensitive filesystems.

Note that all files specified on the command line will be processed regardless of extension unless the -ext option is used.

-F[OFFSET] (-fixBase)
Fix the base for maker notes offsets. A common problem with some image editors is that offsets in the maker notes are not adjusted properly when the file is modified. This may cause the wrong values to be extracted for some maker note entries when reading the edited file. This option allows an integer OFFSET to be specified for adjusting the maker notes base offset. If no OFFSET is given, ExifTool takes its best guess at the correct base. Note that exiftool will automatically fix the offsets for images which store original offset information (eg. newer Canon models). Offsets are fixed permanently if -F is used when writing EXIF to an image. eg)

```
exiftool -F -exif:resolutionunit=inches image.jpg
```

-fast[NUM]
Increase speed of extracting information. With -fast (or -fast1), ExifTool will not scan to the end of a JPEG image to check for an AFCP or PreviewImage trailer, or past the first comment in GIF images or the audio/video data in WAV/AVI files to search for additional metadata. These speed benefits are small when reading images directly from disk, but can be substantial if piping images through a network connection. For more substantial speed benefits, -fast2 also causes exiftool to avoid extracting any EXIF MakerNote information, and to stop processing at the IDAT chunk of PNG images and the mdat atom of QuickTime-format files (but note that some files may store metadata after this). -fast3 avoids extracting metadata from the file, and returns only pseudo System tags, but still reads the file header to obtain an educated guess at FileType. -fast4 doesn't even read the file header, and returns only System tags and a FileType based on the file extension. -fast5 also disables generation of the Composite tags (like -e). Has no effect when writing.

Note that a separate -fast setting may be used for evaluation of a -if condition, or when ordering files with the -fileOrder option. See the -if and -fileOrder options for details.

-fileOrder[NUM] [-]TAG
Set file processing order according to the sorted value of the

specified TAG. For example, to process files in order of date:

```
exiftool -fileOrder DateTimeOriginal DIR
```

Additional **-fileOrder** options may be added for secondary sort keys. Numbers are sorted numerically, and all other values are sorted alphabetically. Files missing the specified tag are sorted last. The sort order may be reversed by prefixing the tag name with a "-" (eg. "-fileOrder -createdate"). Print conversion of the sorted values is disabled with the **-n** option, or a "#" appended to the tag name. Other formatting options (eg. **-d**) have no effect on the sorted values. Note that the **-fileOrder** option can incur large performance penalty since it involves an additional initial processing pass of all files, but this impact may be reduced by specifying a NUM to effectively set the **-fast** level for the initial pass. For example, **-fileOrder4** may be used if TAG is a pseudo System tag. If multiple **-fileOrder** options are used, the extraction is done at the lowest **-fast** level. Note that files are sorted across directory boundaries if multiple input directories are specified.

-i DIR (**-ignore**)

Ignore specified directory name. DIR may be either an individual folder name, or a full path. If a full path is specified, it must match the Directory tag exactly to be ignored. Use multiple **-i** options to ignore more than one directory name. A special DIR value of "SYMLINKS" (case sensitive) may be specified to avoid recursing into directories which are symbolic links when the **-r** option is used. As well, a value of "HIDDEN" (case sensitive) may be used to ignore files with names that start with a "." (ie. hidden files on Unix systems) when scanning a directory.

-if[NUM] EXPR

Specify a condition to be evaluated before processing each FILE. EXPR is a Perl-like logic expression containing tag names prefixed by "$" symbols. It is evaluated with the tags from each FILE in turn, and the file is processed only if the expression returns true. Unlike Perl variable names, tag names are not case sensitive and may contain a hyphen. As well, tag names may have a leading group names separated by colons, and/or a trailing "#" character to disable print conversion. The expression $GROUP:all evaluates to 1 if any tag exists in the specified "GROUP", or 0 otherwise (see note 2 below). When multiple **-if** options are used, all conditions must be satisfied to process the file. Returns an exit status of 2 if all files fail the condition. Below are a few examples:

```
# extract shutterspeed from all Canon images in a directory
exiftool -shutterspeed -if '$make eq "Canon"' dir

# add one hour to all images created on or after Apr. 2, 2006
exiftool -alldates+=1 -if '$CreateDate ge "2006:04:02"' dir

# set EXIF ISO value if possible, unless it is set already
exiftool '-exif:iso<iso' -if 'not $exif:iso' dir

# find images containing a specific keyword (case insensitive)
exiftool -if '$keywords =~ /harvey/i' -filename dir
```

Adding NUM to the **-if** option causes a separate processing pass to be executed for evaluating EXPR at a **-fast** level given by NUM (see the **-fast** option documentation for details). Without NUM, only one processing pass is done at the level specified by the **-fast** option. For example, using **-if5** is possible if EXPR uses only pseudo System tags, and may significantly speed processing if enough files fail the condition.

The expression has access to the current ExifTool object through $self, and the following special functions are available to allow short-circuiting of the file processing. Both functions have a

return value of 1. Case is significant for function names.

```
End()    - end processing after this file
EndDir() - end processing of files in the current directory
           after this file (not compatible with -fileOrder)
```

Notes:

1) The -n and -b options also apply to tags used in EXPR.

2) Some binary data blocks are not extracted unless specified explicitly. These tags are not available for use in the -if condition unless they are also specified on the command line. The alternative is to use the $GROUP:all syntax. (eg. Use $exif:all instead of $exif in EXPR to test for the existence of EXIF tags.)

3) Tags in the string are interpolated in a similar way to -p before the expression is evaluated. In this interpolation, $/ is converted to a newline and $$ represents a single "$" symbol. So Perl variables, if used, require a double "$", and regular expressions ending in $/ must use $$/ instead.

4) The condition may only test tags from the file being processed. To process one file based on tags from another, two steps are required. For example, to process XMP sidecar files in directory "DIR" based on tags from the associated NEF:

```
exiftool -if EXPR -p '$directory/$filename' -ext nef DIR > nef.txt
exiftool -@ nef.txt -srcfile %d%f.xmp ...
```

5) The -a option has no effect on the evaluation of the expression, and the values of duplicate tags are accessible only by specifying a group name (such as a family 4 instance number, eg. $Copy1:TAG, $Copy2:TAG, etc).

6) A special "OK" UserParam is available to test the success of the previous command when -execute was used, and may be used like any other tag in the condition (ie. "$OK").

7) The API RequestTags option is automatically set for all tags used in the -if condition.

-m (**-ignoreMinorErrors**)

Ignore minor errors and warnings. This enables writing to files with minor errors and disables some validation checks which could result in minor warnings. Generally, minor errors/warnings indicate a problem which usually won't result in loss of metadata if ignored. However, there are exceptions, so ExifTool leaves it up to you to make the final decision. Minor errors and warnings are indicated by "[minor]" at the start of the message. Warnings which affect processing when ignored are indicated by "[Minor]" (with a capital "M"). Note that this causes missing values in **-tagsFromFile**, **-p** and **-if** strings to be set to an empty string rather than an undefined value.

-o OUTFILE or FMT (**-out**)

Set the output file or directory name when writing information. Without this option, when any "real" tags are written the original file is renamed to "FILE_original" and output is written to FILE. When writing only FileName and/or Directory "pseudo" tags, **-o** causes the file to be copied instead of moved, but directories specified for either of these tags take precedence over that specified by the **-o** option.

OUTFILE may be "-" to write to stdout. The output file name may also be specified using a FMT string in which %d, %f and %e represent the directory, file name and extension of FILE. Also, %c may be used to add a copy number. See the **-w** option for FMT string examples.

The output file is taken to be a directory name if it already exists as a directory or if the name ends with '/'. Output directories are created if necessary. Existing files will not be overwritten. Combining the **-overwrite_original** option with **-o** causes the original source file to be erased after the output file is successfully written.

A special feature of this option allows the creation of certain types of files from scratch, or with the metadata from another type of file. The following file types may be created using this technique:

XMP, EXIF, EXV, MIE, ICC/ICM, VRD, DR4

The output file type is determined by the extension of OUTFILE (specified as "-.EXT" when writing to stdout). The output file is then created from a combination of information in FILE (as if the **-tagsFromFile** option was used), and tag values assigned on the command line. If no FILE is specified, the output file may be created from scratch using only tags assigned on the command line.

-overwrite_original

Overwrite the original FILE (instead of preserving it by adding "_original" to the file name) when writing information to an image. Caution: This option should only be used if you already have separate backup copies of your image files. The overwrite is implemented by renaming a temporary file to replace the original. This deletes the original file and replaces it with the edited version in a single operation. When combined with **-o**, this option causes the original file to be deleted if the output file was successfully written (ie. the file is moved instead of copied).

-overwrite_original_in_place

Similar to **-overwrite_original** except that an extra step is added to allow the original file attributes to be preserved. For example, on a Mac this causes the original file creation date, type, creator, label color, icon, Finder tags, other extended attributes and hard links to the file to be preserved (but note that the Mac OS resource fork is always preserved unless specifically deleted with "-rsrc:all="). This is implemented by opening the original file in update mode and replacing its data with a copy of a temporary file before deleting the temporary. The extra step results in slower performance, so the **-overwrite_original** option should be used instead unless necessary.

Note that this option reverts to the behaviour of the **-overwrite_original** option when also writing the FileName and/or Directory tags.

-P (-preserve)

Preserve the filesystem modification date/time ("FileModifyDate") of the original file when writing. Note that some filesystems store a creation date (ie. "FileCreateDate" on Windows and Mac systems) which is not affected by this option. This creation date is preserved on Windows systems where Win32API::File and Win32::API are available regardless of this setting. For other systems, the **-overwrite_original_in_place** option may be used if necessary to preserve the creation date. The **-P** option is superseded by any value written to the FileModifyDate tag.

-password PASSWD

Specify password to allow processing of password-protected PDF documents. If a password is required but not given, a warning is issued and the document is not processed. This option is ignored if a password is not required.

-progress[NUM][:[TITLE]]

Show the progress when processing files. Without a colon, the **-progress** option adds a progress count in brackets after the name

of each processed file, giving the current file number and the total number of files to be processed. Implies the **-v0** option, causing the names of processed files to also be printed when writing. When combined with the **-if** option, the total count includes all files before the condition is applied, but files that fail the condition will not have their names printed. If NUM is specified, the progress is shown every NUM input files.

If followed by a colon (ie. **-progress:**), the console window title is set according to the specified TITLE string. If no TITLE is given, a default TITLE string of "ExifTool %p%%" is assumed. In the string, %f represents the file name, %p is the progress as a percent, %r is the progress as a ratio, %##b is a progress bar of width "##" (where "##" is an integer specifying the bar width in characters, or 20 characters by default if "##" is omitted), and %% is a % character. May be combined with the normal **-progress** option to also show the progress count in console messages. (Note: For this feature to function correctly on Mac/Linux, stderr must go to the console.)

-q (**-quiet**)
Quiet processing. One **-q** suppresses normal informational messages, and a second **-q** suppresses warnings as well. Error messages can not be suppressed, although minor errors may be downgraded to warnings with the **-m** option, which may then be suppressed with "-q -q".

-r[.] (**-recurse**)
Recursively process files in subdirectories. Only meaningful if FILE is a directory name. Subdirectories with names beginning with "." are not processed unless "." is added to the option name (ie. **-r.** or **-recurse.**). By default, exiftool will also follow symbolic links to directories if supported by the system, but this may be disabled with "-i SYMLINKS" (see the **-i** option for details). Combine this with **-ext** options to control the types of files processed.

-scanForXMP
Scan all files (even unsupported formats) for XMP information unless found already. When combined with the **-fast** option, only unsupported file types are scanned. Warning: It can be time consuming to scan large files.

-u (**-unknown**)
Extract values of unknown tags. Add another **-u** to also extract unknown information from binary data blocks. This option applies to tags with numerical tag ID's, and causes tag names like "Exif_0xc5d9" to be generated for unknown information. It has no effect on information types which have human-readable tag ID's (such as XMP), since unknown tags are extracted automatically from these formats.

-U (**-unknown2**)
Extract values of unknown tags as well as unknown information from some binary data blocks. This is the same as two **-u** options.

-wm MODE (**-writeMode**)
Set mode for writing/creating tags. MODE is a string of one or more characters from the list below. The default write mode is "wcg".

```
w - Write existing tags
c - Create new tags
g - create new Groups as necessary
```

For example, use "-wm cg" to only create new tags (and avoid editing existing ones).

The level of the group is the SubDirectory level in the metadata structure. For XMP or IPTC this is the full XMP/IPTC block (the

family 0 group), but for EXIF this is the individual IFD (the family 1 group).

-z (-zip)
When reading, causes information to be extracted from .gz and .bz2 compressed images (only one image per archive; requires gzip and bzip2 to be available). When writing, causes compressed information to be written if supported by the metadata format (eg. compressed textual metadata in PNG), disables the recommended padding in embedded XMP (saving 2424 bytes when writing XMP in a file), and writes XMP in shorthand format -- the equivalent of setting the API Compress=1 and Compact="NoPadding,Shorthand".

Other options

-@ ARGFILE
Read command-line arguments from the specified file. The file contains one argument per line (NOT one option per line -- some options require additional arguments, and all arguments must be placed on separate lines). Blank lines and lines beginning with "#" are ignored (unless they start with "#[CSTR]", in which case the rest of the line is treated as a C string, allowing standard C escape sequences such as "\n" for a newline). White space at the start of a line is removed. Normal shell processing of arguments is not performed, which among other things means that arguments should not be quoted and spaces are treated as any other character. ARGFILE may exist relative to either the current directory or the exiftool directory unless an absolute pathname is given.

For example, the following ARGFILE will set the value of Copyright to "Copyright YYYY, Phil Harvey", where "YYYY" is the year of CreateDate:

```
-d
%Y
-copyright<Copyright $createdate, Phil Harvey
```

Arguments in ARGFILE behave exactly the same as if they were entered at the location of the -@ option on the command line, with the exception that the -config and -common_args options may not be used in an ARGFILE.

-k (-pause)
Pause with the message "-- press any key --" or "-- press RETURN --" (depending on your system) before terminating. This option is used to prevent the command window from closing when run as a Windows drag and drop application.

-list, -listw, -listf, -listr, -listwf, -listg[NUM], -listd, -listx
Print a list of all valid tag names (-list), all writable tag names (-listw), all supported file extensions (-listf), all recognized file extensions (-listr), all writable file extensions (-listwf), all tag groups [in a specified family] (-listg[NUM]), all deletable tag groups (-listd), or an XML database of tag details including language translations (-listx). The -list, -listw and -listx options may be followed by an additional argument of the form "-GROUP:All" to list only tags in a specific group, where "GROUP" is one or more family 0-2 group names (excepting EXIF IFD groups) separated by colons. With -listg, NUM may be given to specify the group family, otherwise family 0 is assumed. The -l option may be combined with -listf, -listr or -listwf to add file descriptions to the list. The -lang option may be combined with -listx to output descriptions in a single language. Here are some examples:

```
-list               # list all tag names
-list -EXIF:All     # list all EXIF tags
-list -xmp:time:all # list all XMP tags relating to time
-listw -XMP-dc:All  # list all writable XMP-dc tags
```

```
    -listf              # list all supported file extensions
    -listr              # list all recognized file extensions
    -listwf             # list all writable file extensions
    -listg1             # list all groups in family 1
    -listd              # list all deletable groups
    -listx -EXIF:All    # list database of EXIF tags in XML format
    -listx -XMP:All -s  # list short XML database of XMP tags
```

When combined with **-listx**, the **-s** option shortens the output by omitting the descriptions and values (as in the last example above), and **-f** adds 'flags' and 'struct' attributes if applicable. The flags are formatted as a comma-separated list of the following possible values: Avoid, Binary, List, Mandatory, Permanent, Protected, Unknown and Unsafe (see the Tag Name documentation). For XMP List tags, the list type (Alt, Bag or Seq) is added to the flags, and flattened structure tags are indicated by a Flattened flag with 'struct' giving the ID of the parent structure.

Note that none of the **-list** options require an input FILE.

-ver Print exiftool version number. The **-v** option may be added to print addition system information (see the README file of the full distribution for more details about optional libraries), or **-v2** to also list the Perl include directories.

-- Indicates the end of options. Any remaining arguments are treated as file names, even if they begin with a dash ("-").

Special features

-geotag TRKFILE

Geotag images from the specified GPS track log file. Using the **-geotag** option is equivalent to writing a value to the "Geotag" tag. The GPS position is interpolated from the track at a time specified by the value written to the "Geotime" tag. If "Geotime" is not specified, the value is copied from "DateTimeOriginal#" (the "#" is added to copy the unformatted value, avoiding potential conflicts with the **-d** option). For example, the following two commands are equivalent:

```
exiftool -geotag trk.log image.jpg
exiftool -geotag trk.log "-Geotime<DateTimeOriginal#" image.jpg
```

When the "Geotime" value is converted to UTC, the local system timezone is assumed unless the date/time value contains a timezone. Writing "Geotime" causes the following tags to be written (provided they can be calculated from the track log, and they are supported by the destination metadata format): GPSLatitude, GPSLatitudeRef, GPSLongitude, GPSLongitudeRef, GPSAltitude, GPSAltitudeRef, GPSDateStamp, GPSTimeStamp, GPSDateTime, GPSTrack, GPSTrackRef, GPSSpeed, GPSSpeedRef, GPSImgDirection, GPSImgDirectionRef, GPSPitch, GPSRoll, AmbientTemperature and CameraElevationAngle. By default, tags are created in EXIF, and updated in XMP only if they already exist. However, "EXIF:Geotime" or "XMP:Geotime" may be specified to write only EXIF or XMP tags respectively. Note that GPSPitch and GPSRoll are non-standard, and require user-defined tags in order to be written.

The "Geosync" tag may be used to specify a time correction which is applied to each "Geotime" value for synchronization with GPS time. For example, the following command compensates for image times which are 1 minute and 20 seconds behind GPS:

```
exiftool -geosync=+1:20 -geotag a.log DIR
```

Advanced "Geosync" features allow a piecewise linear time drift correction and synchronization from previously geotagged images. See "geotag.html" in the full ExifTool distribution for more information.

Multiple -geotag options may be used to concatenate GPS track log data. Also, a single -geotag option may be used to load multiple track log files by using wildcards in the TRKFILE name, but note that in this case TRKFILE must be quoted on most systems (with the notable exception of Windows) to prevent filename expansion. For example:

```
exiftool -geotag "TRACKDIR/*.log" IMAGEDIR
```

Currently supported track file formats are GPX, NMEA RMC/GGA/GLL, KML, IGC, Garmin XML and TCX, Magellan PMGNTRK, Honeywell PTNTHPR, Bramor gEO, Winplus Beacon TXT, and GPS/IMU CSV files. See "GEOTAGGING EXAMPLES" for examples. Also see "geotag.html" in the full ExifTool distribution and the Image::ExifTool Options for more details and for information about geotag configuration options.

-globalTimeShift SHIFT

Shift all formatted date/time values by the specified amount when reading. Does not apply to unformatted (-n) output. SHIFT takes the same form as the date/time shift when writing (see Image::ExifTool::Shift.pl for details), with a negative shift being indicated with a minus sign ("-") at the start of the SHIFT string. For example:

```
# return all date/times, shifted back by 1 hour
exiftool -globalTimeShift -1 -time:all a.jpg

# set the file name from the shifted CreateDate (-1 day) for
# all images in a directory
exiftool "-filename<createdate" -globaltimeshift "-0:0:1 0:0:0" \
    -d %Y%m%d-%H%M%S.%%e dir
```

-use MODULE

Add features from specified plug-in MODULE. Currently, the MWG module is the only plug-in module distributed with exiftool. This module adds read/write support for tags as recommended by the Metadata Working Group. As a convenience, "-use MWG" is assumed if the group name prefix starts with "MWG:" exactly for any requested tag. See the MWG Tags documentation for more details. Note that this option is not reversible, and remains in effect until the application terminates, even across the -execute option.

Utilities

-restore_original
-delete_original[!]

These utility options automate the maintenance of the "_original" files created by exiftool. They have no effect on files without an "_original" copy. The -restore_original option restores the specified files from their original copies by renaming the "_original" files to replace the edited versions. For example, the following command restores the originals of all JPG images in directory "DIR":

```
exiftool -restore_original -ext jpg DIR
```

The -delete_original option deletes the "_original" copies of all files specified on the command line. Without a trailing "!" this option prompts for confirmation before continuing. For example, the following command deletes "a.jpg_original" if it exists, after asking "Are you sure?":

```
exiftool -delete_original a.jpg
```

These options may not be used with other options to read or write tag values in the same command, but may be combined with options such -ext, -if, -r, -q and -v.

Advanced options

Among other things, the advanced options allow complex processing to be performed from a single command without the need for additional scripting. This may be particularly useful for implementations such as Windows drag-and-drop applications. These options may also be used to improve performance in multi-pass processing by reducing the overhead required to load exiftool for each invocation.

-api OPT[[^]=[VAL]]
: Set ExifTool API option. OPT is an API option name. The option value is set to 1 if =VAL is omitted. If VAL is omitted, the option value is set to undef if "=" is used, or an empty string with "^=". See Image::ExifTool Options for a list of available API options. This overrides API options set via the config file.

-common_args
: Specifies that all arguments following this option are common to all executed commands when **-execute** is used. This and the **-config** option are the only options that may not be used inside a **-@** ARGFILE. Note that by definition this option and its arguments MUST come after all other options on the command line.

-config CFGFILE
: Load specified configuration file instead of the default ".ExifTool_config". If used, this option must come before all other arguments on the command line and applies to all **-execute**'d commands. The CFGFILE must exist relative to the current working directory or the exiftool application directory unless an absolute path is specified. Loading of the default config file may be disabled by setting CFGFILE to an empty string (ie. ""). See <https://exiftool.org/config.html> and config_files/example.config in the full ExifTool distribution for details about the configuration file syntax.

-echo[NUM] TEXT
: Echo TEXT to stdout (**-echo** or **-echo1**) or stderr (**-echo2**). Text is output as the command line is parsed, before the processing of any input files. NUM may also be 3 or 4 to output text (to stdout or stderr respectively) after processing is complete. For **-echo3** and **-echo4**, "${status}" may be used in the TEXT string to represent the numerical exit status of the command (see "EXIT STATUS").

-efile[NUM][!] ERRFILE
: Save the names of files giving errors (NUM missing or 1), files that were unchanged (NUM is 2), files that fail the **-if** condition (NUM is 4), or any combination thereof by summing NUM (eg. **-efile3** is the same has having both **-efile** and **-efile2** options with the same ERRFILE). By default, file names are appended to any existing ERRFILE, but ERRFILE is overwritten if an exclamation point is added to the option (eg. **-efile!**). Saves the name of the file specified by the **-srcfile** option if applicable.

-execute[NUM]
: Execute command for all arguments up to this point on the command line (plus any arguments specified by **-common_args**). The result is as if the commands were executed as separate command lines (with the exception of the **-config** and **-use** options which remain in effect for subsequent commands). Allows multiple commands to be executed from a single command line. NUM is an optional number that is echoed in the "{ready}" message when using the **-stay_open** feature. If a NUM is specified, the **-q** option no longer suppresses the output "{readyNUM}" message.

-fileNUM ALTFILE
: Read tags from an alternate source file. These tags are accessed via the family 8 group name (eg. "File1:TAG", "File2:TAG", etc). ALTFILE may contain filename formatting codes %d, %f and %e. Among other things, this allows tags from different files to be compared and combined using the **-if** and **-p** options.

-list_dir
List directories themselves instead of their contents. This option effectively causes directories to be treated as normal files when reading and writing. For example, with this option the output of the "ls -la" command on Mac/Linux may be approximated by this exiftool command:

```
exiftool -list_dir -T -ls-l -api systemtags -fast5 .* *
```

(The **-T** option formats the output in tab-separated columns, **-ls-l** is a shortcut tag, the API SystemTags option is required to extract some necessary tags, and the **-fast5** option is added for speed since only system tags are being extracted.)

-srcfile FMT
Specify a different source file to be processed based on the name of the original FILE. This may be useful in some special situations for processing related preview images or sidecar files. See the **-w** option for a description of the FMT syntax. Note that file name FMT strings for all options are based on the original FILE specified from the command line, not the name of the source file specified by **-srcfile**.

For example, to copy metadata from NEF files to the corresponding JPG previews in a directory where other JPG images may exist:

```
exiftool -ext nef -tagsfromfile @ -srcfile %d%f.jpg dir
```

If more than one **-srcfile** option is specified, the files are tested in order and the first existing source file is processed. If none of the source files already exist, then exiftool uses the first **-srcfile** specified.

A FMT of "@" may be used to represent the original FILE, which may be useful when specifying multiple **-srcfile** options (eg. to fall back to processing the original FILE if no sidecar exists).

When this option is used, two special UserParam tags (OriginalFileName and OriginalDirectory) are generated to allow access to the original FILE name and directory.

-stay_open FLAG
If FLAG is 1 or "True" (case insensitive), causes exiftool keep reading from the **-@** ARGFILE even after reaching the end of file. This feature allows calling applications to pre-load exiftool, thus avoiding the overhead of loading exiftool for each command. The procedure is as follows:

1) Execute "exiftool -stay_open True -@ ARGFILE", where ARGFILE is the name of an existing (possibly empty) argument file or "-" to pipe arguments from the standard input.

2) Write exiftool command-line arguments to ARGFILE, one argument per line (see the **-@** option for details).

3) Write "-execute\n" to ARGFILE, where "\n" represents a newline sequence. (Note: You may need to flush your write buffers here if using buffered output.) ExifTool will then execute the command with the arguments received up to this point, send a "{ready}" message to stdout when done (unless the **-q** or **-T** option is used), and continue trying to read arguments for the next command from ARGFILE. To aid in command/response synchronization, any number appended to the **-execute** option is echoed in the "{ready}" message. For example, "-execute613" results in "{ready613}". When this number is added, **-q** no longer suppresses the "{ready}" message. (Also, see the **-echo3** and **-echo4** options for additional ways to pass signals back to your application.)

4) Repeat steps 2 and 3 for each command.

5) Write "-stay_open\nFalse\n" (or "-stay_open\n0\n") to ARGFILE when done. This will cause exiftool to process any remaining command-line arguments then exit normally.

The input ARGFILE may be changed at any time before step 5 above by writing the following lines to the currently open ARGFILE:

```
-stay_open
True
-@
NEWARGFILE
```

This causes ARGFILE to be closed, and NEWARGFILE to be kept open. (Without the **-stay_open** here, exiftool would have returned to reading arguments from ARGFILE after reaching the end of NEWARGFILE.)

Note: When writing arguments to a disk file there is a delay of up to 0.01 seconds after writing "-execute\n" before exiftool starts processing the command. This delay may be avoided by sending a CONT signal to the exiftool process immediately after writing "-execute\n". (There is no associated delay when writing arguments via a pipe with "-@ -", so the signal is not necessary when using this technique.)

-userParam PARAM[[^]=[VAL]]

Set user parameter. PARAM is an arbitrary user parameter name. This is an interface to the API UserParam option (see the Image::ExifTool Options documentation), and provides a method to access user-defined parameters in arguments to the **-if** and **-p** options as if they were any other tag. Appending a hash tag ("#") to PARAM (eg. "-userParam MyTag#=yes") also causes the parameter to be extracted as a normal tag in the UserParam group. Similar to the **-api** option, the parameter value is set to 1 if =VAL is omitted, undef if just VAL is omitted with "=", or an empty string if VAL is omitted with "^=".

```
exiftool -p '$test from $filename' -userparam test=Hello FILE
```

Advanced formatting feature

An advanced formatting feature allows modification of the value of any tag interpolated within a **-if** or **-p** option argument, or a **-tagsFromFile** redirection string. Tag names within these strings are prefixed by a "$" symbol, and an arbitrary Perl expression may be applied to the tag value by placing braces around the tag name and inserting the expression after the name, separated by a semicolon (ie. "${TAG;EXPR}"). The expression acts on the value of the tag through the default input variable ($_), and has access to the full ExifTool API through the current ExifTool object ($self) and the tag key ($tag). It may contain any valid Perl code, including translation ("tr///") and substitution ("s///") operations, but note that braces within the expression must be balanced. The example below prints the camera Make with spaces translated to underlines, and multiple consecutive underlines replaced by a single underline:

```
exiftool -p '${make;tr/ /_/;s/__+/_/g}' image.jpg
```

An "@" may be added after the tag name to make the expression act on individual list items for list-type tags, simplifying list processing. Set $_ to undef to remove an item from the list. As an example, the following command returns all subjects not containing the string "xxx":

```
exiftool -p '${subject@;$_=undef if /xxx/}' image.jpg
```

A default expression of "tr(/\\?*:|"<>\0)()d" is assumed if the expression is empty (ie. "${TAG;}"). This removes the characters / \ ? * : | < > and null from the printed value. (These characters are illegal in Windows file names, so this feature is useful if tag values

are used in file names.)

Helper functions

"DateFmt"

Simplifies reformatting of individual date/time values. This function acts on a standard EXIF-formatted date/time value in $_ and formats it according to the specified format string (see the -d option). To avoid trying to reformat an already-formatted date/time value, a "#" must be added to the tag name (as in the example below) if the -d option is also used. For example:

```
exiftool -p '${createdate#;DateFmt("%Y-%m-%d_%H%M%S")}' a.jpg
```

"ShiftTime"

Shifts EXIF-formatted date/time string by a specified amount. Start with a leading minus sign to shift backwards in time. See Image::ExifTool::Shift.pl for details about shift syntax. For example, to shift a date/time value back by one year:

```
exiftool -p '${createdate;ShiftTime("-1:0:0 0")}' a.jpg
```

"NoDups"

Removes duplicate items from a list with a separator specified by the -sep option. This function is most useful when copying list-type tags. For example, the following command may be used to remove duplicate Keywords:

```
exiftool -sep '##' '-keywords<${keywords;NoDups}' a.jpg
```

The -sep option is necessary to split the string back into individual list items when writing to a list-type tag.

An optional flag argument may be set to 1 to cause "NoDups" to set $_ to undef if no duplicates existed, thus preventing the file from being rewritten unnecessarily:

```
exiftool -sep '##' '-keywords<${keywords;NoDups(1)}' a.jpg
```

Note that function names are case sensitive.

WINDOWS UNICODE FILE NAMES

In Windows, command-line arguments are specified using the current code page and are recoded automatically to the system code page. This recoding is not done for arguments in ExifTool arg files, so by default filenames in arg files use the system code page. Unfortunately, these code pages are not complete character sets, so not all file names may be represented.

ExifTool 9.79 and later allow the file name encoding to be specified with "-charset filename=CHARSET", where "CHARSET" is the name of a valid ExifTool character set, preferably "UTF8" (see the -charset option for a complete list). Setting this triggers the use of Windows wide-character i/o routines, thus providing support for most Unicode file names (see note 4). But note that it is not trivial to pass properly encoded file names on the Windows command line (see <https://exiftool.org/faq.html#Q18> for details), so placing them in a UTF-8 encoded -@ argfile and using "-charset filename=utf8" is recommended if possible.

A warning is issued if a specified filename contains special characters and the filename character set was not provided. However, the warning may be disabled by setting "-charset filename=""", and ExifTool may still function correctly if the system code page matches the character set used for the file names.

When a directory name is provided, the file name encoding need not be

specified (unless the directory name contains special characters), and ExifTool will automatically use wide-character routines to scan the directory.

The filename character set applies to the FILE arguments as well as filename arguments of **-@**, **-geotag**, **-o**, **-p**, **-srcfile**, **-tagsFromFile**, **-csv=**, **-j=** and **-TAG<=**. However, it does not apply to the **-config** filename, which always uses the system character set. The "-charset filename=" option must come before the **-@** option to be effective, but the order doesn't matter with respect to other options.

Notes:

1) FileName and Directory tag values still use the same encoding as other tag values, and are converted to/from the filename character set when writing/reading if specified.

2) Unicode support is not yet implemented for other Windows-based systems like Cygwin.

3) See "WRITING READ-ONLY FILES" below for a note about editing read-only files with Unicode names.

4) Unicode file names with surrogate pairs (code points over U+FFFF) still cause problems.

WRITING READ-ONLY FILES

In general, ExifTool may be used to write metadata to read-only files provided that the user has write permission in the directory. However, there are three cases where file write permission is also required:

1) When using the **-overwrite_original_in_place** option.

2) When writing only pseudo System tags (eg. FileModifyDate).

3) On Windows if the file has Unicode characters in its name, and a) the **-overwrite_original** option is used, or b) the "_original" backup already exists.

Hidden files in Windows behave as read-only files when attempting to write any real tags to the file -- an error is generated when using the **-overwrite_original_in_place**, otherwise writing should be successful and the hidden attribute will be removed. But the **-if** option may be used to avoid processing hidden files (provided Win32API::File is available):

```
exiftool -if "$fileattributes !~ /Hidden/" ...
```

READING EXAMPLES

Note: Beware when cutting and pasting these examples into your terminal! Some characters such as single and double quotes and hyphens may have been changed into similar-looking yet functionally-different characters by the text formatter used to display this documentation. Also note that Windows users must use double quotes instead of single quotes as below around arguments containing special characters.

```
exiftool -a -u -g1 a.jpg
```

Print all meta information in an image, including duplicate and unknown tags, sorted by group (for family 1). For performance reasons, this command may not extract all available metadata. (Metadata in embedded documents, metadata extracted by external utilities, and metadata requiring excessive processing time may not be extracted). Add "-ee3" and "-api RequestAll=3" to the command to extract absolutely everything available.

```
exiftool -common dir
```

Print common meta information for all images in "dir". "-common" is a shortcut tag representing common EXIF meta information.

```
exiftool -T -createdate -aperture -shutterspeed -iso dir > out.txt
```

List specified meta information in tab-delimited column form for all images in "dir" to an output text file named "out.txt".

exiftool -s -ImageSize -ExposureTime b.jpg
Print ImageSize and ExposureTime tag names and values.

exiftool -l -canon c.jpg d.jpg
Print standard Canon information from two image files.

exiftool -r -w .txt -common pictures
Recursively extract common meta information from files in "pictures" directory, writing text output to ".txt" files with the same names.

exiftool -b -ThumbnailImage image.jpg > thumbnail.jpg
Save thumbnail image from "image.jpg" to a file called "thumbnail.jpg".

exiftool -b -JpgFromRaw -w _JFR.JPG -ext NEF -r .
Recursively extract JPG image from all Nikon NEF files in the current directory, adding "_JFR.JPG" for the name of the output JPG files.

exiftool -a -b -W %d%f_%t%-c.%s -preview:all dir
Extract all types of preview images (ThumbnailImage, PreviewImage, JpgFromRaw, etc.) from files in directory "dir", adding the tag name to the output preview image file names.

exiftool -d '%r %a, %B %e, %Y' -DateTimeOriginal -S -s -ext jpg .
Print formatted date/time for all JPG files in the current directory.

exiftool -IFD1:XResolution -IFD1:YResolution image.jpg
Extract image resolution from EXIF IFD1 information (thumbnail image IFD).

exiftool '-*resolution*' image.jpg
Extract all tags with names containing the word "Resolution" from an image.

exiftool -xmp:author:all -a image.jpg
Extract all author-related XMP information from an image.

exiftool -xmp -b a.jpg > out.xmp
Extract complete XMP data record intact from "a.jpg" and write it to "out.xmp" using the special "XMP" tag (see the Extra tags in Image::ExifTool::TagNames).

exiftool -p '$filename has date $dateTimeOriginal' -q -f dir
Print one line of output containing the file name and DateTimeOriginal for each image in directory "dir".

exiftool -ee3 -p '$gpslatitude, $gpslongitude, $gpstimestamp' a.m2ts
Extract all GPS positions from an AVCHD video.

exiftool -icc_profile -b -w icc image.jpg
Save complete ICC_Profile from an image to an output file with the same name and an extension of ".icc".

exiftool -htmldump -w tmp/%f_%e.html t/images
Generate HTML pages from a hex dump of EXIF information in all images from the "t/images" directory. The output HTML files are written to the "tmp" directory (which is created if it didn't exist), with names of the form 'FILENAME_EXT.html'.

exiftool -a -b -ee -embeddedimage -W Image_%.3g3.%s file.pdf
Extract embedded JPG and JP2 images from a PDF file. The output images will have file names like "Image_#.jpg" or "Image_#.jp2", where "#" is the ExifTool family 3 embedded document number for the image.

WRITING EXAMPLES

Note that quotes are necessary around arguments which contain certain special characters such as ">", "<" or any white space. These quoting techniques are shell dependent, but the examples below will work for most Unix shells. With the Windows cmd shell however, double quotes should be used (eg. -Comment="This is a new comment").

exiftool -Comment='This is a new comment' dst.jpg
Write new comment to a JPG image (replaces any existing comment).

exiftool -comment= -o newdir -ext jpg .
Remove comment from all JPG images in the current directory, writing the modified images to a new directory.

exiftool -keywords=EXIF -keywords=editor dst.jpg
Replace existing keyword list with two new keywords ("EXIF" and "editor").

exiftool -Keywords+=word -o newfile.jpg src.jpg
Copy a source image to a new file, and add a keyword ("word") to the current list of keywords.

exiftool -exposurecompensation+=-0.5 a.jpg
Decrement the value of ExposureCompensation by 0.5 EV. Note that += with a negative value is used for decrementing because the -= operator is used for conditional deletion (see next example).

exiftool -credit-=xxx dir
Delete Credit information from all files in a directory where the Credit value was "xxx".

exiftool -xmp:description-de='kühl' -E dst.jpg
Write alternate language for XMP:Description, using HTML character escaping to input special characters.

exiftool -all= dst.jpg
Delete all meta information from an image. Note: You should NOT do this to RAW images (except DNG) since proprietary RAW image formats often contain information in the makernotes that is necessary for converting the image.

exiftool -all= -comment='lonely' dst.jpg
Delete all meta information from an image and add a comment back in. (Note that the order is important: "-comment='lonely' -all=" would also delete the new comment.)

exiftool -all= --jfif:all dst.jpg
Delete all meta information except JFIF group from an image.

exiftool -Photoshop:All= dst.jpg
Delete Photoshop meta information from an image (note that the Photoshop information also includes IPTC).

exiftool -r -XMP-crss:all= DIR
Recursively delete all XMP-crss information from images in a directory.

exiftool '-ThumbnailImage<=thumb.jpg' dst.jpg
Set the thumbnail image from specified file (Note: The quotes are necessary to prevent shell redirection).

exiftool '-JpgFromRaw<=%d%f_JFR.JPG' -ext NEF -r .
Recursively write JPEG images with filenames ending in "_JFR.JPG" to the JpgFromRaw tag of like-named files with extension ".NEF" in the current directory. (This is the inverse of the "-JpgFromRaw" command of the "READING EXAMPLES" section above.)

exiftool -DateTimeOriginal-='0:0:0 1:30:0' dir
Adjust original date/time of all images in directory "dir" by

subtracting one hour and 30 minutes. (This is equivalent to "-DateTimeOriginal-=1.5". See Image::ExifTool::Shift.pl for details.)

exiftool -createdate+=3 -modifydate+=3 a.jpg b.jpg
Add 3 hours to the CreateDate and ModifyDate timestamps of two images.

exiftool -AllDates+=1:30 -if '$make eq "Canon"' dir
Shift the values of DateTimeOriginal, CreateDate and ModifyDate forward by 1 hour and 30 minutes for all Canon images in a directory. (The AllDates tag is provided as a shortcut for these three tags, allowing them to be accessed via a single tag.)

exiftool -xmp:city=Kingston image1.jpg image2.nef
Write a tag to the XMP group of two images. (Without the "xmp:" this tag would get written to the IPTC group since "City" exists in both, and IPTC is preferred by default.)

exiftool -LightSource-='Unknown (0)' dst.tiff
Delete "LightSource" tag only if it is unknown with a value of 0.

exiftool -whitebalance-=auto -WhiteBalance=tung dst.jpg
Set "WhiteBalance" to "Tungsten" only if it was previously "Auto".

exiftool -comment-= -comment='new comment' a.jpg
Write a new comment only if the image doesn't have one already.

exiftool -o %d%f.xmp dir
Create XMP meta information data files for all images in "dir".

exiftool -o test.xmp -owner=Phil -title='XMP File'
Create an XMP data file only from tags defined on the command line.

exiftool '-ICC_Profile<=%d%f.icc' image.jpg
Write ICC_Profile to an image from a ".icc" file of the same name.

exiftool -hierarchicalkeywords='{keyword=one,children={keyword=B}}'
Write structured XMP information. See <https://exiftool.org/struct.html> for more details.

exiftool -trailer:all= image.jpg
Delete any trailer found after the end of image (EOI) in a JPEG file. A number of digital cameras store a large PreviewImage after the JPEG EOI, and the file size may be reduced significantly by deleting this trailer. See the JPEG Tags documentation for a list of recognized JPEG trailers.

COPYING EXAMPLES

These examples demonstrate the ability to copy tag values between files.

exiftool -tagsFromFile src.cr2 dst.jpg
Copy the values of all writable tags from "src.cr2" to "dst.jpg", writing the information to same-named tags in the preferred groups.

exiftool -TagsFromFile src.jpg -all:all dst.jpg
Copy the values of all writable tags from "src.jpg" to "dst.jpg", preserving the original tag groups.

exiftool -all= -tagsfromfile src.jpg -exif:all dst.jpg
Erase all meta information from "dst.jpg" image, then copy EXIF tags from "src.jpg".

exiftool -exif:all= -tagsfromfile @ -all:all -unsafe bad.jpg
Rebuild all EXIF meta information from scratch in an image. This technique can be used in JPEG images to repair corrupted EXIF information which otherwise could not be written due to errors.

The "Unsafe" tag is a shortcut for unsafe EXIF tags in JPEG images which are not normally copied. See the tag name documentation for more details about unsafe tags.

exiftool -Tagsfromfile a.jpg out.xmp
Copy meta information from "a.jpg" to an XMP data file. If the XMP data file "out.xmp" already exists, it will be updated with the new information. Otherwise the XMP data file will be created. Only metadata-only files may be created like this (files containing images may be edited but not created). See "WRITING EXAMPLES" above for another technique to generate XMP files.

exiftool -tagsFromFile a.jpg -XMP:All= -ThumbnailImage= -m b.jpg
Copy all meta information from "a.jpg" to "b.jpg", deleting all XMP information and the thumbnail image from the destination.

exiftool -TagsFromFile src.jpg -title -author=Phil dst.jpg
Copy title from one image to another and set a new author name.

exiftool -TagsFromFile a.jpg -ISO -TagsFromFile b.jpg -comment dst.jpg
Copy ISO from one image and Comment from another image to a destination image.

exiftool -tagsfromfile src.jpg -exif:all --subifd:all dst.jpg
Copy only the EXIF information from one image to another, excluding SubIFD tags.

exiftool '-FileModifyDate<DateTimeOriginal' dir
Use the original date from the meta information to set the same file's filesystem modification date for all images in a directory. (Note that "-TagsFromFile @" is assumed if no other **-TagsFromFile** is specified when redirecting information as in this example.)

exiftool -TagsFromFile src.jpg '-xmp:all<all' dst.jpg
Copy all possible information from "src.jpg" and write in XMP format to "dst.jpg".

exiftool '-Description<${FileName;s/\.[^.]*$//}' dir
Set the image Description from the file name after removing the extension. This example uses the "Advanced formatting feature" to perform a substitution operation to remove the last dot and subsequent characters from the file name.

exiftool -@ iptc2xmp.args -iptc:all= a.jpg
Translate IPTC information to XMP with appropriate tag name conversions, and delete the original IPTC information from an image. This example uses iptc2xmp.args, which is a file included with the ExifTool distribution that contains the required arguments to convert IPTC information to XMP format. Also included with the distribution are xmp2iptc.args (which performs the inverse conversion) and a few more .args files for other conversions between EXIF, IPTC and XMP.

exiftool -tagsfromfile %d%f.CR2 -r -ext JPG dir
Recursively rewrite all "JPG" images in "dir" with information copied from the corresponding "CR2" images in the same directories.

exiftool '-keywords+<make' image.jpg
Add camera make to list of keywords.

exiftool '-comment<ISO=$exif:iso Exposure=${shutterspeed}' dir
Set the Comment tag of all images in "dir" from the values of the EXIF:ISO and ShutterSpeed tags. The resulting comment will be in the form "ISO=100 Exposure=1/60".

exiftool -TagsFromFile src.jpg -icc_profile dst.jpg
Copy ICC_Profile from one image to another.

exiftool -TagsFromFile src.jpg -all:all dst.mie

Copy all meta information in its original form from a JPEG image to a MIE file. The MIE file will be created if it doesn't exist. This technique can be used to store the metadata of an image so it can be inserted back into the image (with the inverse command) later in a workflow.

exiftool -o dst.mie -all:all src.jpg
This command performs exactly the same task as the command above, except that the **-o** option will not write to an output file that already exists.

exiftool -b -jpgfromraw -w %d%f_%ue.jpg -execute -b -previewimage -w %d%f_%ue.jpg -execute -tagsfromfile @ -srcfile %d%f_%ue.jpg -overwrite_original -common_args --ext jpg DIR
[Advanced] Extract JpgFromRaw or PreviewImage from all but JPG files in DIR, saving them with file names like "image_EXT.jpg", then add all meta information from the original files to the extracted images. Here, the command line is broken into three sections (separated by **-execute** options), and each is executed as if it were a separate command. The **-common_args** option causes the "--ext jpg DIR" arguments to be applied to all three commands, and the **-srcfile** option allows the extracted JPG image to be the source file for the third command (whereas the RAW files are the source files for the other two commands).

RENAMING EXAMPLES

By writing the "FileName" and "Directory" tags, files are renamed and/or moved to new directories. This can be particularly useful and powerful for organizing files by date when combined with the **-d** option. New directories are created as necessary, but existing files will not be overwritten. The format codes %d, %f and %e may be used in the new file name to represent the directory, name and extension of the original file, and %c may be used to add a copy number if the file already exists (see the **-w** option for details). Note that if used within a date format string, an extra '%' must be added to pass these codes through the date/time parser. (And further note that in a Windows batch file, all '%' characters must also be escaped, so in this extreme case '%%%%f' is necessary to pass a simple '%f' through the two levels of parsing.) See <https://exiftool.org/filename.html> for additional documentation and examples.

exiftool -filename=new.jpg dir/old.jpg
Rename "old.jpg" to "new.jpg" in directory "dir".

exiftool -directory=%e dir
Move all files from directory "dir" into directories named by the original file extensions.

exiftool '-Directory<DateTimeOriginal' -d %Y/%m/%d dir
Move all files in "dir" into a directory hierarchy based on year, month and day of "DateTimeOriginal". eg) This command would move the file "dir/image.jpg" with a "DateTimeOriginal" of "2005:10:12 16:05:56" to "2005/10/12/image.jpg".

exiftool -o . '-Directory<DateTimeOriginal' -d %Y/%m/%d dir
Same effect as above except files are copied instead of moved.

exiftool '-filename<%f_${model;}.%e' dir
Rename all files in "dir" by adding the camera model name to the file name. The semicolon after the tag name inside the braces causes characters which are invalid in Windows file names to be deleted from the tag value (see the "Advanced formatting feature" for an explanation).

exiftool '-FileName<CreateDate' -d %Y%m%d_%H%M%S%%-c.%%e dir
Rename all images in "dir" according to the "CreateDate" date and time, adding a copy number with leading '-' if the file already exists ("%-c"), and preserving the original file extension (%e). Note the extra '%' necessary to escape the filename codes (%c and %e) in the date format string.

exiftool -r '-FileName<CreateDate' -d %Y-%m-%d/%H%M_%%f.%%e dir
Both the directory and the filename may be changed together via the "FileName" tag if the new "FileName" contains a '/'. The example above recursively renames all images in a directory by adding a "CreateDate" timestamp to the start of the filename, then moves them into new directories named by date.

exiftool '-FileName<${CreateDate}_$filenumber.jpg' -d %Y%m%d -ext jpg .
Set the filename of all JPG images in the current directory from the CreateDate and FileNumber tags, in the form "20060507_118-1861.jpg".

GEOTAGGING EXAMPLES

ExifTool implements geotagging via 3 special tags: Geotag (which for convenience is also implemented as an exiftool option), Geosync and Geotime. The examples below highlight some geotagging features. See <https://exiftool.org/geotag.html> for additional documentation.

exiftool -geotag track.log a.jpg
Geotag an image ("a.jpg") from position information in a GPS track log ("track.log"). Since the "Geotime" tag is not specified, the value of DateTimeOriginal is used for geotagging. Local system time is assumed unless DateTimeOriginal contains a timezone.

exiftool -geotag t.log -geotime='2009:04:02 13:41:12-05:00' a.jpg
Geotag an image with the GPS position for a specific time.

exiftool -geotag log.gpx '-xmp:geotime<createdate' dir
Geotag all images in directory "dir" with XMP tags instead of EXIF tags, based on the image CreateDate.

exiftool -geotag a.log -geosync=-20 dir
Geotag images in directory "dir", accounting for image timestamps which were 20 seconds ahead of GPS.

exiftool -geotag a.log -geosync=1.jpg -geosync=2.jpg dir
Geotag images using time synchronization from two previously geotagged images (1.jpg and 2.jpg), synchronizing the image and GPS times using a linear time drift correction.

exiftool -geotag a.log '-geotime<${createdate}+01:00' dir
Geotag images in "dir" using CreateDate with the specified timezone. If CreateDate already contained a timezone, then the timezone specified on the command line is ignored.

exiftool -geotag= a.jpg
Delete GPS tags which may have been added by the geotag feature. Note that this does not remove all GPS tags -- to do this instead use "-gps:all=".

exiftool -xmp:geotag= a.jpg
Delete XMP GPS tags which were added by the geotag feature.

exiftool -xmp:geotag=track.log a.jpg
Geotag an image with XMP tags, using the time from DateTimeOriginal.

exiftool -geotag a.log -geotag b.log -r dir
Combine multiple track logs and geotag an entire directory tree of images.

exiftool -geotag 'tracks/*.log' -r dir
Read all track logs from the "tracks" directory.

exiftool -p gpx.fmt dir > out.gpx
Generate a GPX track log from all images in directory "dir". This example uses the "gpx.fmt" file included in the full ExifTool distribution package and assumes that the images in "dir" have all been previously geotagged.

PIPING EXAMPLES

cat a.jpg | exiftool -
Extract information from stdin.

exiftool image.jpg -thumbnailimage -b | exiftool -
Extract information from an embedded thumbnail image.

cat a.jpg | exiftool -iptc:keywords+=fantastic - > b.jpg
Add an IPTC keyword in a pipeline, saving output to a new file.

curl -s http://a.domain.com/bigfile.jpg | exiftool -fast -
Extract information from an image over the internet using the cURL utility. The **-fast** option prevents exiftool from scanning for trailer information, so only the meta information header is transferred.

exiftool a.jpg -thumbnailimage -b | exiftool -comment=wow - | exiftool a.jpg -thumbnailimage'<=-'
Add a comment to an embedded thumbnail image. (Why anyone would want to do this I don't know, but I've included this as an example to illustrate the flexibility of ExifTool.)

INTERRUPTING EXIFTOOL

Interrupting exiftool with a CTRL-C or SIGINT will not result in partially written files or temporary files remaining on the hard disk. The exiftool application traps SIGINT and defers it until the end of critical processes if necessary, then does a proper cleanup before exiting.

EXIT STATUS

The exiftool application exits with a status of 0 on success, or 1 if an error occurred, or 2 if all files failed the **-if** condition (for any of the commands if **-execute** was used).

AUTHOR

SEE ALSO

Image::ExifTool(3pm), Image::ExifTool::TagNames(3pm), Image::ExifTool::Shortcuts(3pm), Image::ExifTool::Shift.pl