

NAME

exiftool – Read and write meta information in files

SYNOPSIS

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

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

exiftool [*OPTIONS*] **-TagsFromFile** *SRCFILE* [*-SRCTAG[>DSTTAG]...*] *FILE...*

exiftool [*-ver* | *-list[w|f|g[*NUM*]|d]*]

For specific examples, see the EXAMPLES sections below.

DESCRIPTION

A command-line interface to Image::ExifTool, used for reading and writing meta information in image, audio and video files. *FILE* is a source file name, directory name, or *-* for the standard input. Information is read from the source file and output in readable form to the console (or written to an output text file with the *-w* option).

To write or copy information, new values are specified with the *-TAG=[VALUE]* syntax or the **-TagsFromFile** option. This causes *FILE* to be rewritten, and by default the original file is preserved with *_original* appended to the file name. (Be sure to verify that the new file is OK before erasing the original.)

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

File Types						Meta Information	
-----						-----	
3FR	r	ITC	r	PPM	r/w	EXIF	r/w/c
ACR	r	JNG	r/w	PPT	r	GPS	r/w/c
AI	r	JP2	r/w	PS	r/w	IPTC	r/w/c
AIFF	r	JPEG	r/w	PSD	r/w	XMP	r/w/c
APE	r	K25	r	QTIF	r	MakerNotes	r/w/c
ARW	r	KDC	r	RA	r	Photoshop IRB	r/w/c
ASF	r	M4A	r	RAF	r/w	ICC Profile	r/w/c
AVI	r	MEF	r/w	RAM	r	MIE	r/w/c
BMP	r	MIE	r/w/c	RAW	r/w	JFIF	r/w/c
BTF	r	MIFF	r	RIFF	r	Ducky APP12	r/w/c
CR2	r/w	MNG	r/w	RM	r	CIFF	r/w
CRW	r/w	MOS	r/w	SR2	r	AFCP	r/w
CS1	r/w	MOV	r	SRF	r	JPEG 2000	r
DCM	r	MP3	r	SVG	r	DICOM	r
DCR	r	MP4	r	SWF	r	Flash	r
DIVX	r	MPC	r	THM	r/w	FlashPix	r
DNG	r/w	MPG	r	TIFF	r/w	QuickTime	r
DOC	r	MRW	r/w	VRD	r/w/c	GeoTIFF	r
EPS	r/w	NEF	r/w	WAV	r	PrintIM	r
ERF	r/w	OGG	r	WDP	r/w	ID3	r
FLAC	r	ORF	r/w	WMA	r	Kodak Meta	r
FLV	r	PBM	r/w	WMV	r	Ricoh RMETA	r
FPX	r	PDF	r/w	X3F	r	Picture Info	r
GIF	r/w	PEF	r/w	XLS	r	Adobe APP14	r
HDP	r/w	PGM	r/w	XMP	r/w/c	APE	r
HTML	r	PICT	r			Vorbis	r
ICC	r/w/c	PNG	r/w			(and more)	

Note: If *FILE* is a directory name, then only file types with recognized extensions are processed when reading, and only writable types are written. However, the *-ext* option may be used to force processing of

files with any extension.

OPTIONS

Case is not significant for any command-line option (including tag and group names), except for single-character options if the corresponding upper case option is defined. Multiple options may NOT be combined into a single argument, because that would be interpreted as a tag name.

`-TAG`

Extract information for specified tag (ie. `-CreateDate`). See `Image::ExifTool::TagNames` for documentation on available tag names. A tag name may include a leading group name separated by a colon (ie. `-XMP:Creator`). Any family 0 or 1 group name may be used, and the group name may be prefixed by a digit to specify family number (ie. `-1IPTC:Keywords`). Use the `-listg` or `-listg1` option to list valid family 0 or 1 group names.

A special tag name of `All` may be used to indicate all meta information. This is particularly useful when a group name is specified to extract all information in a group. (`*` is a synonym for `All`, but must be quoted if used on the command line to prevent shell globbing.)

If no tags are specified, all available information is extracted.

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. May also be used following a `-TagsFromFile` option to exclude tags from being copied, or to exclude groups from being deleted when deleting all information (ie. `-all= --exif:all` deletes all but EXIF information).

Note that this will not exclude individual tags from a group delete. Instead, the tags must be recovered using the `-TagsFromFile` option.

`-TAG[+]=VALUE`

Write a new value for the specified tag (ie. `-comment=wow`), or delete the tag if no *VALUE* is given (ie. `-comment=`). `+=` and `-=` are used to add or remove existing entries from a list, or to shift date/time values (see `Image::ExifTool::Shift.pl` for details), and `-=` may be used to conditionally remove or replace a tag (see "WRITING EXAMPLES" for examples).

If a group name is not specified for *TAG*, then the information is written to the preferred group, which is the first group in the following list where *TAG* is valid: 1) EXIF, 2) IPTC, 3) XMP, 4) MakerNotes.

The special `All` tag may be used in this syntax only if a *VALUE* is NOT given. This causes all meta information to be deleted (or all information in a group if `-GROUP:All=` is used). Note that not all groups are deletable. Also, within an image some groups may be contained within others, and these groups are removed if the super group is deleted. Below are lists of these group dependencies:

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.

`-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. `+<=` or `-<=` may also be used to add or delete specific list

entries, or to shift date/time values.

-@ *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 which must be placed on separate lines). Blank lines and lines beginning with # and are ignored. Normal shell processing of arguments is not performed, which among other things means that arguments should not be quoted. *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
```

-a Allow duplicate tag names in the output. Without this option, duplicates are suppressed.

-b Output requested data in **binary** format without tag names or descriptions. This option is mainly used for extracting embedded images, but it may also be useful in separating values of list-type tags since a newline is used instead of a comma to separate these values in the **-b** output.

-c *FMT*

Set the print format for GPS coordinates. *FMT* uses the same syntax as the `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" (the default)
"%d deg %.4f min"	54 deg 59.3800 min
"% .6f degrees"	54.989667 degrees

-d *FMT*

Set the format for **date/time** tag values. Consult `strftime` man page for *FMT* syntax. The default format is equivalent to “%Y:%m:%d %H:%M:%S”. This option has no effect on date-only or time-only tags, ignores timezone information if present, and currently doesn’t apply when writing.

-D Show tag ID number in **Decimal**.

-e Print existing tags only — don’t calculate composite tags.

-E Escape characters in output values for HTML. Implied with the **-h** option.

-ext *EXT* (or **--ext** *EXT*)

Process only files with (**-ext**) or without (**--ext**) a specified extension. There may be multiple **-ext** and **--ext** options. Extensions may begin with a leading ‘.’, and case is not significant. For example:

```
exiftool -ext .JPG DIR # process only JPG files
exiftool --ext crw --ext dng DIR # process all but CRW and DNG
exiftool --ext . DIR # ignore if no extension
```

-f Force printing of tags even if their values are not found.

-F[*OFFSET*]

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 (ie. newer Canon models). Offsets are fixed

permanently if used when writing an image.

-fast

Increase speed of extracting information from JPEG images. With this option, ExifTool will not scan to the end of a JPEG image to check for an AFPC or PreviewImage trailer. The speed benefits are small when reading images directly from disk, but can be substantial if piping images through a network connection.

-g[*NUM*]

Organize output by tag **group**. *NUM* specifies the group family number, and may be 0 (general location), 1 (specific location) or 2 (category). If not specified, **-g0** is assumed. Use the **-listg** option to list all group names for a specified family.

-G[*NUM*]

Same as **-g** but print **Group** name for each tag.

-h Use **HTML** formatting for output. Implies **-E** option.

-H Show tag ID number in **Hexadecimal**.

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

-i *DIR*

Ignore specified directory name. May be multiple **-i** options.

-if *EXPR*

Specify a condition to be evaluated before processing each *FILE*. *EXPR* is a Perl-like expression containing tag names prefixed by \$ symbols. It is evaluated with the tags from each *FILE* in turn, and processing proceeds only if the expression returns true. Unlike Perl variable names, tag names are not case sensitive and may contain a hyphen or a leading group name separated by a colon. When multiple **-if** options are used, all conditions must be satisfied to process the file. Below are a few examples:

```
# extract shutter speed 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
```

-k Pause with the message `-- press any key --` before terminating. This option is used to prevent the command window from closing when run as a Windows drag and drop application.

-l Use long 2-line Canon-style output format.

-L Convert 16-bit Unicode characters in output to Windows Latin1 (cp1252) instead of the default UTF-8. When writing, **-L** is used to specify that input text values are Latin1 instead of UTF-8.

-list, -listw, -listf, -listg[*NUM*], -listd

Print a **list** of all valid tag names (**-list**), all writable tag names (**-listw**), all recognized file extensions (**-listf**), all tag groups [in a specified family] (**-listg[*NUM*]**), or all deletable tag groups (**-listd**). The **-list** and **-listw** options may be followed by an additional argument of the form **-GROUP:All** to list all tags in a specific group. With **-listg**, *NUM* may be given to specify the

group family, otherwise family 0 is assumed. For example:

```
-list           # list all tag names
-list -EXIF:All # list all EXIF tags
-listw -XMP-dc:All # list all writable XMP-dc tags
-listf         # list all recognized file extensions
-listgl       # list all groups in family 1
-listd       # list all deletable groups
```

-m Ignore **minor** errors. Allows writing if some minor errors occur, or extraction of embedded images that aren't in standard JPG format.

-n Read and write values as **numbers** instead of words. This option disables the print conversion that is applied when extracting values to make them more readable, and the inverse print conversion when writing. For example:

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

and the following two writing commands have the same effect

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

-o *OUTFILE* or *FMT*

Set the **output** file or directory name when writing information. (Without this option, the original file is renamed to `FILE_original` and output is sent to `FILE`.) 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 it to be used to create certain types of files from scratch. Currently, this can only be done with XMP and ICC/ICM files. The file is created from a combination of information in *FILE* and tag values assigned on the command line. This is done by specifying a file extension of `'.XMP'`, `'.ICC'` or `'.ICM'` for *OUTFILE*. The output file may be created even if no *FILE* is specified, provided some appropriate tag values are specified 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.

-overwrite_original_in_place

Similar to the **-overwrite_original** option except that an extra step is added to allow the original file attributes to be preserved. On a Macintosh for example, this preserves the original file type, creator and icon. This is implemented by copying the contents of a temporary file over the original, then deleting the temporary file. The extra step results in slower performance, so the **-overwrite_original** option should be used instead unless necessary.

-p *FMTFILE* or *STR*

Print output in the format specified by the given file or string (and ignore other format options). Tag names in the format file or string begin with a `$` symbol and may contain an optional group name. Case is not significant. Braces `{ }` may be used around the tag name to separate it from subsequent text. Use `$$` to represent a `$` symbol. `$/` may be used for a newline. In the file, lines beginning

with # are ignored. For example, this format file:

```
# this is a comment line
File $FileName was created on $DateTimeOriginal
(f/$Aperture, ${ShutterSpeed}s, ISO $EXIF:ISO)
```

produces output like this:

```
File test.jpg was created on 2003:10:31 15:44:19
(f/5.6, 1/60s, ISO 100)
```

If a tag does not exist, the output value is set to '-' if the **-f** option is used. Otherwise a minor warning is issued and the line with the missing tag is not printed. However, if the **-m** option is used, minor warnings are ignored and the line is printed with an empty tag value.

- P** Preserve date/time of original file (`FileModifyDate`) when writing.
- q** 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.
- r** Recursively scan subdirectories. Only meaningful if *FILE* is a directory name.
- s** Print tag names instead of descriptions. This is the short output format. Add up to 3 **-s** options for even shorter formats:

```
-s          - print tag names instead of descriptions
-s -s      - no extra spaces to column-align values
-s -s -s   - print values only
```

Also effective when combined with **-t** or **-h** options.

- S** Very short format. The same as two **-s** options. Extra spaces used to column-align values are not printed.
- ScanForXMP**
Scan all files (even unrecognized formats) for XMP information unless found already. When combined with the **-fast** option, only unrecognized file types are scanned. Warning: It can be time consuming to scan large files.
- t** 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.
- T** Output tag values in Table form. (Equivalent to **-t -S -q -f**.)
- 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. If no tags are specified, then all tags from the source file are copied. More than one **-TagsFromFile** option may be used to copy tags from multiple files.

By default, this option will commute information between same-named tags in different groups and write each tag to the preferred group. This allows information to be automatically translated when copying between images of different formats. However, if a group name is specified for a tag then the information is written to the original group (unless redirected to another group, see below). This works even if `All` is used as a group name, so `-All:All` is used to specify that all information be copied to the same group in the destination file.

SRCFILE may be the same as *FILE* to move information around within a 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 (see note 3 below). 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*. See **-w** option for *FMT* string examples.

A powerful redirection feature allows a destination tag to be specified for each extracted tag. With this feature, information may be written to a tag with a different name or group. This is done using `"-SRCTAG>DSTTAG"` on the command line after `-TagsFromFile` (`"-DSTTAG<SRCTAG"` also works). Note that this argument must be quoted to prevent shell redirection, and there is no = sign as when assigning new values. Both source and destination tags may be prefixed by a group name, and All or * may be used as a tag or group name. If no destination group is specified, the information is written to the preferred group. 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"`.

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 for more details about this syntax. Strings starting with a = sign must insert a single space after the < to avoid confusion with the <= syntax which would otherwise attempt to set 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 is preserved.

See “COPYING EXAMPLES” for examples using `-TagsFromFile`.

Notes:

- 1) 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.
- 2) 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. Also, since the PreviewImage referenced from the maker notes may be rather large, it is not copied, and must be transferred separately if desired.
- 3) When performing complex batch processing, it is important to note that the order of operations is different for tags copied in batch mode. In general, tags are copied from batch-mode files after all other command-line arguments have been applied. (The exception is that a group delete is always performed last if there are no subsequent tag assignments on the command line.) For example, the following two commands are not equivalent:

```
# (not batch mode): Sets xmp:title to 'NEW'
exiftool -tagsfromfile a.jpg -xmp:title -xmp:title=NEW a.jpg

# (batch mode): Preserves original title if it exists
exiftool -tagsfromfile @ -xmp:title -xmp:title=NEW a.jpg
```

- `-u` Extract values of **unknown** tags. Add another `-u` to also extract unknown information from binary data blocks.
- `-U` Extract values of **unknown** tags as well as unknown information from binary data blocks. This is the same as two `-u` options.
- `-v[NUM]`

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. This option suppresses normal console output unless specific tags are being extracted. `-v0` is used when writing to print only the processed file names, and the “Nothing changed” message for files that were not modified.

- `-ver` Print version number and exit.

`-w` *EXT* or *FMT*

Write console output to a file with name ending in *EXT* 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 didn't 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 overwritten, and output directories are created automatically if necessary.

Note: In a Windows BAT file the '%' character is represented by "%%", so an argument like "%d%f.txt" is written as "%%d%f.txt".

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

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, a copy number of zero is omitted, 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 ...
```

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 the copy number (%c), the numbers are changed to an alphabetical base (see example H above). Also, %c may be modified by 'n' to count using natural numbers starting from 1, instead of 0 (see example F).

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

-x TAG

Exclude the specified tag. There may be multiple **-x** options. This has the same effect as **---TAG** on the command line. May also be used following a **-TagsFromFile** option to exclude tags from being copied.

-z When reading, causes information to be extracted from .gz and .bz2 compressed images. (Only one image per archive.) When writing, causes compressed information to be written if supported by the image format. (ie. The PNG format supports compressed text.)

Advanced Options

The following options allow complex processing to be performed with 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.

-common_args

Specifies that all arguments following this option are common to all executed commands when **-execute** is used. This is the only option that may not be used inside a **-@ ARGFILE**.

-execute

Execute command for all arguments up to this point on the command line. Allows multiple commands to be executed from a single command line.

-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 name specified on the command line, not the name of the source file specified by **-srcfile**.

READING EXAMPLES

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

Print all meta information in an image, including duplicate and unknown tags, sorted by group (for family 1).

```
exiftool -common dir
```

Print common meta information for all images in *dir*.

```
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 into files with the same names but with a *.txt* extension.

```
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 -b -ThumbnailImage image.jpg > thumbnail.jpg
```

Save thumbnail image from *image.jpg* to a file called *thumbnail.jpg*.

```
exiftool -b -PreviewImage 118_1834.JPG > preview.jpg
```

Extract preview image from JPG file and write it to *preview.jpg*.

```
exiftool -b -JpgFromRaw -w _JFR.JPG -ext CRW -r .
```

Recursively extract JPG image from all Canon CRW files in the current directory, adding *_JFR.JPG* for the name of the output JPG files.

```
exiftool -d '%r %a, %B %e, %Y' -DateTimeOriginal -S -s *.jpg
```

Print formatted date/time for all JPG files in a directory.

```
exiftool -IFD1:XRResolution -IFD1:YResolution
```

Extract image resolution from EXIF IFD1 information (thumbnail image IFD).

```
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 -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'`.

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 (ie. `-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 *.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 -credit=-xxx dir`
 Delete Credit information from all files in a directory where the Credit value was (`xxx`).

`exiftool -all= dst.jpg`
 Delete all meta information from an 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= --jif: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 CRW -r .`
 Recursively write JPEG images with filenames ending in `_JFR.JPG` to the `JpgFromRaw` tag of like-named files with extension `.CRW` 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 -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.

COPYING EXAMPLES

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

`exiftool -TagsFromFile src.crw dst.jpg`
 Copy the values of all writable tags from `src.crw` to `dst.jpg`, writing the information to 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 -tagsfromfile a.jpg out.xmp`
 Copy meta information `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 XMP, ICC and MIE files may be created like this (other file types 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 '-DateTimeOriginal>FileModifyDate' 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 '-all>xmp:all' dst.jpg`
 Copy all possible information from `src.jpg` and write in XMP format to `dst.jpg`.

```
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 is `xmp2iptc.args`, which performs the inverse conversion.

```
exiftool -tagsfromfile %d%f.CRW -r -ext JPG dir
```

Recursively rewrite all JPG images in `dir` with information copied from the corresponding CRW images in the same directories.

```
exiftool '-make+>keywords' 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 -if '$jpgfromraw' -b -jpgfromraw -w %d%f_%ue.jpg -execute -if '$previewimage' -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.)

```
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`. ie) 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 '-filename<%f_${focallength}.'%e' dir
```

Rename all files in `dir` by adding `FocalLength` to the file name.

```
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<${CreateDate}_$filename.jpg' -d %Y%m%d *.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”.

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.

```
wget -qO - http://a.domain.com/bigfile.jpg | exiftool -fast -
```

Extract information from an image over the internet using the GNU `wget` 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`.)

BUGS

`ExifTool` does not handle information stored in the resource fork on Macintosh filesystems.

AUTHOR

Copyright 2003–2008, Phil Harvey

This is free software; you can redistribute it and/or modify it under the same terms as Perl itself.

SEE ALSO

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