EXIFutils

Image Metadata Utilities

Command Reference Guide

for
Linux & MacOS X

V3.0

Copyright Notice

Trademark Acknowledgements

All terms or logos used in this manual that are known to be registered trademarks, trademarks, or service marks of entities other than the author are listed below:

Microsoft® Windows 98®, Microsoft® Windows 95®, Microsoft® Windows® 2000, Microsoft® Windows® XP, and Windows® Vista are trademarks and/or service marks or registered trademarks of Microsoft Corporation or its subsidiaries in the United States and/or in other countries.

Mac OS X is a trademark of Apple Computer, Inc., registered in the U.S. and other countries. The EXIFutils Getting Started Guide is an independent publication and has not been authorized, sponsored, or otherwise approved by Apple Computer, Inc.

Linux is a registered trademark of Linus Torvalds.

Fedora® and the Infinity design logo are trademarks of Red Hat, Inc.

UNIX is the property of The Open Group.

Photoshop is a trademark of Adobe Systems Incorporated.

Flashpix is a trademark of Digital Imaging Group, Inc.

All other trademarks or service marks are the property of their respective owners.

Contents

1 Introduction

1.1 Structure of this manual

The sections in this manual cover the following topics:

Section 1 "Introduction" provides an overview of this manual and the changes that have been made since the previous version of EXIFutils.

Section 2 "Basic Command Syntax" explains the command option formats that are common to all EXIFutils commands.

Section 4 "Referencing Fields Using Nicknames" describes methods used to refer to Meta data fields.

Section 5 "Working with Dates and Times" explains the EXIFutils features that assist in manipulating date and time fields.

Section 6 "Character Sets" explains how EXIFutils interprets multinational characters, and how EXIFutils can be configured to interpret and display different character sets.

Section 7 "Templates" explains the use of substitution *templates* in EXIFutils commands.

Section 8 "Command Reference" describes in detail the features and use of each EXIFutils command.

1.2 Intended Audience

This manual is intended for users of the EXIFutils who understand the basic concepts of the EXIFutils and want detailed information on the use of each command. It assumes that the reader has a working knowledge of the use of Linux/UNIX systems. In particular it is assumed that the reader understands:

- The use of at least one UNIX command shell,
- How to redirect the output of a command into a file,
- How to pipe the output of one command into another command,
- The use of quotes and wildcards on the command line.

1.3 Conventions Used in this Manual

The following conventions are used in this manual.

`myphoto.jpg`

Text in the `Courier New` font represents examples that must be typed exactly as shown.

`field-list.txt`

Text in *`Italicized Courier New`* font represents text for which you must substitute an appropriate value, for example in the following code fragment:

```
-u “your name”
```

you should replace *`“your name”`* with your own name.

`[ ]`

Square brackets surrounding text in a command description indicate that the text is optional. For example, if a command is shown as having the following option:

```
[-t]
```

then the `-t` option is optional and need not be entered.

`[a|b]`

Indicates that you must enter either a or b. For example, if a command is shown as having the option:

```
-t [a|r]
```

then you must enter either

```
-t a
```

or

```
-t r
```

`...`

The ellipse symbol “…” indicates that the item preceding it can be repeated multiple of times. The example below indicates that multiple file names can be entered separated by commas:

```
filename,...
```

`\`

A backslash character at the end of a line indicates that whatever follows on the next line is a continuation of the same command. Example:

```
exiflist -o l -f make,model \
my-file.jpg
```

is the same as entering:

```
exiflist -o l -f make,model my-file.jpg
```

1.4 Feedback

If you have any feedback on the content or structure of this manual, or if you have any suggestions on how the EXIFutils can be improved, please send email to: support@hugsan.com.

1.5 What's New in V3.0?

1.5.1 New Features

This section describes the new features introduced in V3.0, and changes to existing features.

- Added support for MacOS X on Intel based platforms
- Add support for Adobe XMP meta data, including:
    - Reading, adding and removing XMP fields
    - Assigning values from legacy EXIF and IPTC metadata fields to their XMP equivalents, and vice versa. See Section 3.5 "Syncronisation between XMP and non-XMP Data Fields".
    - Assigning multiple language string values to fields defined as "language alternate" fields. See Section 4.2.3 "Selecting/Setting Entries in a Language Alternate List".
    - Setting a default language other than English. See Section 4.2.3.2 "EXIFutils Default Language".

  Refer to the following sections for more information:
    - 3.4 "XMP Fields"
    - 4.2 "Nicknames for XMP Structure and List Fields"
- Added support for the following file types:
    - Adobe Digital Negative (`.DNG`) - read/write
    - Olympus Raw (`.ORF`) – read only
    - Sony Raw (`.ARW`) – read only
- Improvments to `exiflist` command.
    - Added new display options:
        - The '`-o dn`' option displays the nicknames and values of all fields present in the image file in an easy to read format (See Section 8.7.2.1 "Full Field Display").
        - The '`-o x`' display the serialized XMP datas block stored in the image file.
    - Simplified `exiflist` by removing the following options:
        - `-l` (display field definitions)
        - `-i` (display character set settings).

      The functionality provided by these options have been moved to the new `exifinfo` command.
    - The comma/tab separated list output format (`-o l`, `-o ln`, and `-o lh` options) now allows you to list all fields present in a file. Previously only fields for which EXIFutils had a definition could be listed.
    - Can now list the values of EXIF fields for which EXIFutils does not have a definition. Previously only fields which were known to EXIFutils could be listed. These "unknown" fields are referenced by using their tag number, eg `tag-01ad` (See Section 4.3 "Selecting EXIF Fields Using "Tag" Value")
    - Change behaviour of `exiflist -o ln` and `-o lf` option. Any selected fields that are not present in the image file will be shown as having a null value. Previously the field would have been ommitted from the output.
- Added `exifinfo` command to display:
    - Meta data field definitions
    - EXIFutils settings, environment variables, and default values
    - The names of known XMP Schemas
- Remove the limit on the number of files that can be edited in a single directory (previously limited to 1000)
- Improved installation on Windows:
    - "Check for Update" option added to allow for checking for new versions of EXIFutils.
    - More reliable setting of PATH environment variable during installation.
- Updated many Maker Note field definitions to include fields/values from newer cameras

1.5.2 Bug Fixes

- Fix problems in the display of GPS values:
  - Seconds values in latitudes and longitudes were incorrectly being ommitted even when minutes values were whole numbers.
  - Leading zeroes were missing from some date/time values.
- Fix bug in `exifdate` when subtracting 59 minutes from a minute value of 00
- Fix bug in setting of IPTC string field values. A null byte was incorrectly being appended to values.
- Remove “buf_p" debugging message that was incorrectly output to stdout after a "Maker Note prefix is incorrect" warning was issued.

1.5.3 Backward Compatibility

Some EXIFutils features have changed in a manner that makes them incompatible with previous version. Thse changes are described in the table below.

Table 1 Feature Backward Compatibility

<table>
<tr><th>Feature Change</th><th>Reason for Change</th><th>Action required</th></tr>
<tr>
<td>In V2.x square brackets [] were used to delimit nicknames to be substituted into field values.
In V3 curly brackets {} (braces) are used.</td>
<td>To avoid confusion when referencing XMP list fields that by XMP convention uses square brackets to delimit list references (see Section 4.2.2).</td>
<td>Change all occurrences in of [] in substitution values and templates to {}.

EXIFutils V2.x command:

<code>exifedit -a description=[ip-headline] myfile.jpg</code>

Equivalent EXIFutils V3 command:

<code>exifedit -a description={ip-headline} myfile.jpg</code></td>
</tr>
<tr>
<td>The following <code>exiflist</code> options have been removed and their functions moved to other options.

<code>-o a (list all metadata)</code>
<code>-o e (list EXIF metadata)</code>
<code>-o i (list IPTC metadata)</code>
<code>-o r (list RMETA metadata)</code></td>
<td>These V2.x <code>-o</code> option values were logically inconsistent with other <code>-o</code> options values as they were used to select the fields to be displayed. All other <code>-o</code> options set the format of the output display and use the <code>-f</code> option to select the fields to be displayed.</td>
<td>Change all uses of these options to their equivalent V3 options as shown below:

<table>
<tr><th>V2.x</th><th>V3.0</th></tr>
<tr><td><code>-o a</code></td><td><code>-o d -f all</code></td></tr>
<tr><td><code>-o e</code></td><td><code>-o d -f exif</code></td></tr>
<tr><td><code>-o i</code></td><td><code>-o d -f iptc</code></td></tr>
<tr><td><code>-o r</code></td><td><code>-o d -f rmeta</code></td></tr>
</table>

EXIFutils V2.x command:

<code>exiflist -o e myfile.jpg</code>

Equivalent EXIFutils V3 command:

<code>exiflist -o d -f exif myfile.jpg</code>

Note that as <code>-o d</code> is the default output format, the following is also equivalent to the above commands:

<code>exiflist -f exif myfile.jpg</code></td>
</tr>
<tr>
<td>In V2.x if you attempted to assign a empty value to a field, then <code>exifedit</code></td>
<td>This was done to make it easier to copy fields between different types of</td>
<td></td>
</tr>
</table>

<table>
<tr><th></th><th></th><th></th></tr>
<tr>
<td>would either set the field to an empty string (if it was a legal value), or report an error if an empty string was not a legal value.
In V3 if you attempt to set a field to an empty value then exifedit will ignore the set operation (i.e. the value will not be changed).</td>
<td>metadata.
For example, if you wanted to copy the EXIF description field into the IPTC ip-headline field you can do this in exifedit with the following exifedit action:
-a description=[ip-headline]
If the ip-headline field is not present in the file, then under V2 this would have set added an EXIF description field and set its value to an empty string. Using V3 theEXIF description field will not be created, which is most often what is required in this situation.</td>
<td></td>
</tr>
<tr>
<td>The exiflist -l (list field definitions) and -i (list characterset settings) options have been removed and their functions included in the new exifinfo command.</td>
<td>The exiflist command’s primary function is to extract/display metadata from image files. Displaying field definitions and characterset settings were a secondary function which is more logically handled by a separate command.</td>
<td>Change all uses of these exiflist options to their exifinfo equivalent V3 options as shown below. See Section 8.5 “EXIFINFO” for a complete description of the exifinfo command.
<table><tr><th>V2.x</th><th>V3.0</th></tr><tr><td>exiflist -l n</td><td>exifinfo -b</td></tr><tr><td>exiflist -l <i>fieldlist</i></td><td>exifinfo -f <i>fieldlist</i></td></tr></table>
EXIFutils V2.x command:
exiflist -l ip-caption
Equivalent EXIFutils V3 command:
exifinfo -f ip-caption</td>
</tr>
</table>

2 Basic Command Syntax

This section provides a brief overview of the command formats common to all EXIFutils commands. These guidelines are correct when using the Windows Command prompt, and when using UNIX command shells based on the Bourne shell. With other command shells and scripting languages the format of the commands may vary. Refer to the documentation provided with your command shell for more information.

The basic format of all commands is:

```
commandname option option option filename filename
```

`commandname` is the name of the EXIFutils command being executed, eg: `exiflist`.

`option` is an option that changes the behaviour of the command. There may be no options, one option, or multiple options. All options start with a ‘`-`’ character.

`filename` is the name of a file on which the command will operate. Depending on the command there may no *`filenames`*, one *`filename`*, or multiple *`filenames`*. For some EXIFutils commands/options the *`filename`* can also be the name of a directory, in which case all files in that directory will be processed.

2.1 Types of Command Options

Command *options* can take one of two formats:

- **Simple options:** consist of a ‘`-`’ followed by a single character that identifies the option being used. The presence of the option affects the behaviour of the command. For example, the ‘`-q`’ option on `exiflist` sets ‘quiet mode’ to suppress warning messages.
- **Complex options:** Like simple options, complex option start with a ‘`-`’ followed by a single character that identifies the option. However complex options are followed by an *option string* that further controls the effect of the option. For example, the ‘`-f`’ options on `exiflist` specifies a list of names of the fields to be displayed. The ‘`-f`’ is therefore followed by a string containing the field list, eg:

  ```
  exiflist -f description,make,model myfile.jpg
  ```

  If the option string contains any of the following characters – `“ ,;[]”` – then the whole option string must be surrounded by quotes:

  ```
  exiflist -f “description, make, model” myfile.jpg
  ```

  ```
  exifedit -a “comment=I took this photo” myfile.jpg
  ```

  If the option string contains quote characters, then each quote must be preceded by the escape character “\” so that the command does not confuse this quote as marking the end of the option string:

  ```
  exifedit -a “comment=\”I took this photo, too\”, shutter=2” myfile.jpg
  ```

3 Types of Metadata

3.1 Introduction

Meta data is information that describes an image. It can include information store in the image file by the camera or scanning device that created the image, as well as information added later to assist in the cataloguing or distribution of the image file.

There are several standards for the format and content of Meta data stored in image files. EXIFutils supports five Meta data formats:

- EXIF
- IPTC
- XMP
- Flashpix
- Ricoh RMETA
- Proprietary formats specific to the type of camera used.

Of these, EXIF, IPTC, and XMP are the man general purpose metadata formats.

EXIF (EXchangeable Image file Format) is a standard created by Japanese Electronic Industry Development Association (JEIDA) for the storage of camera and image information. As such it focuses on storing technical information about the image, but also allows for some descriptive and editorial information to be added (eg: a description of the scene, copyright information).

The IPTC format was developed by the International Press Telecommunications Council to provide a standard way of storing news images. As such is main focus is descriptive, cataloging, and distribution information rather than technical information about the image.

There is a degree of overlap between the data stored in EXIF and IPTC formats. For example, both formats allow for a description of the image, and for a copyright statement, and as a result the same information may be duplicated in the EXIF and IPTC formats.

XMP is a relatively new standard introduced by Adobe to replace the numerous metadata formats described above with a single unified format. XMP is an extensible format that allows third parties such as JEIDA and the IPTC to define how the information they required is stored in XMP format.

At present many images contain a combination of EXIF, IPTC, and XMP formatted meta data, with some data duplicated in all three formats for backward compatibility. See Section 3.5 “Syncronisation between XMP and non-XMP Data Fields” for more information about the relationship between EXIF, IPTC, and XMP formatted fields.

3.2 EXIF Fields

3.2.1 Main and Thumbnail Images

The EXIF Specification allows two versions of an image to be present in a JPEG file. The first, called the *main image*, is the fully resolution image that is the main content of the file. The second, called a *thumbnail image*, is a reduced resolution copy of the main image. The thumbnail image is used by applications reading the JPEG file to quickly display the image without reading the full resolution main image. The thumbnail image is optional.

Some file format specifications (eg, TIFF, DNG) allows for many images to be present in a single file. These include multiple main images, thumbnails, and transparency mask images. The EXIFutils will correctly display details of all images in a file, however if the same EXIF field is present in more than one image, only the first occurrence of each field in the TIFF file can be explicitly referenced (see Section 4 “Referencing Fields Using Nicknames”).

3.2.2 EXIF Fields Storage (non-XMP)

EXIF data consists of a collection of fields. Each field has a *tag*, which identifies the field, and one or more data *values*. A field tag is a four digit hexadecimal number (eg A101), but for ease of use the EXIFutils also assign a *nickname* that can be used to uniquely identify each field. For example, the Exposure Time field (tag 829a) has a nickname of `exp-time`. See Section 4 “Referencing Fields Using Nicknames” for more information about nicknames. A complete list of field nicknames is provided in the EXIFutils Field Reference Guide.

The field value can contain a variety of different types of information, and may have more than one value. For example, the value of the Reference Black and White field (nickname `ref-bw`) consists of six numbers. The types of data values that can be present in a field are shown in Table 2 “Field Data Types”[1].

Table 2 Field Data Types

Data Type	Description
Integer	A positive or negative whole number (eg 20).
Rational Number	A Positive or negative rational number, (eg 2.8)
Text String	A text string. For some fields the string may be any length up to a maximum of 2000 characters[2], while other fields are limited to a specific length.
Defined Value	The field can contain one of a defined set of field values. eg the `flash` field can have a value of “Fired” or “Not Fired”
Date/Time	A date/time in the format YYYY:MM:DD HH:MM:SS. eg 2001:03:16 10:58:01
Special	A range of values that must conform to a set of rules specific to that field. For example, the shutter speed field can must be specified as either a whole number of seconds (eg “30”) or as a fractional number specified as a numerator and denominator (eg. “1/125”)
Binary Data	One or more bytes of binary data, respresented as a hexadedcimal string.

3.2.2.1 EXIF Image File Directories (IFDs)

Related EXIF fields are grouped into collections called Image Field Directories (IFDs). **Table 3** lists the types of IFDs defined in the EXIF specification.

Not all IFDs will be present in all images, and not all IFDs will contain all the possible fields that can be included within that type of IFD. Each camera manufacturer chooses the set of fields each of their camera models will include. The fields recognized by the EXIFutils are listed in the EXIFutils Field Reference Guide grouped by IFD.

Table 3 IFD Types

IFD	Description
Camera	Information about the camera that took the photo, and the picture taking conditions at the time the picture was taken
Image	Camera settings used to take the image
Thumbnail	Information about the thumbnail image
Interoperability	Information about the version of the EXIF Specification used to create the image file
GPS	Information about the location and orientation of the camera at the time the image was taken (using the satellite Global Positioning System)
Maker Note	Information specific to the make and model of camera used

[1] The EXIF Specification defines thirteen data types. The EXIFutils provide a simplified view of these data types to the user.

[2] This limitation is imposed by the EXIFutils, not by the EXIF Specification. When running an unregistered version of EXIFutils, the maximum length to which a string can be set is 99.

The Maker Note is unique among the IFDs in that the EXIF Specification does not define its format. Each camera manufacturer develops its own format for the Maker Note field, and they do not always publish this format. Through experimentation, individuals on the Internet have “reverse engineered” many of the formats used. All Maker Note information provided by the EXIFutils is based on these unofficial sources, and therefore the correctness and completeness of this information cannot be guaranteed. Table 4 lists the camera makes for which EXIFutils provides some level of interpretation of the Maker Note fields. The EXIFutils Field Reference Guide lists the Maker Note fields that EXIFutils supports.

Table 4 Camera Makes for which the Maker Note is Interpreted

Manufacturer
Canon
Casio
Epson
Fujifilm
Minolta / Konica Minolta
Nikon
Olympus
Pentax
Panasonic
Ricoh
Sony

3.3 IPTC Fields

IPTC fields provide information about the content and origin of the image, identification of the copyright holder, and information to assist in cataloguing and distribution of the image. The fields are grouped into *datasets*, which are shown in Table 5.

Table 5 IPTC Field Datasets

Dataset Number	Description
1	Addressing
2	Editorial Information
3	Newsphoto Parameters
6	Abstract Relationship
7	Subfile Size
8	Subfile Data
9	Confirmed Object Size

Most IPTC fields can appear only once in an image file. There are two exceptions: the keyword field (`ip-keyword`) and Supplemental Categories field (`ip-supcat`) can be repeated multiple times, limited only by the amount of available memory.

The EXIFutils Field Reference Guide lists the IPTC fields supported by EXIFutils, including an indication of the fields that can be edited by EXIFutils.

3.4 XMP Fields

The XMP specification provides an extensible framework for storing a wide variety of metadata. Third parties such as camera manufacturers can define new fields to be stored in XMP format. These fields are defined in “schemas”. EXIFutils contains predefined lists of the fields defined in a number of common schemas. See EXIFutils Field Reference Guide for a list of known fields.

The following sections provide a simplified overview of XMP concepts. For a complete description of XMP see the XMP Specification (http://www.adobe.com/products/xmp/).

3.4.1 XMP Field Types

Table 6 lists the basic field types in which XMP metadata can be stored. These are described in more detail in the sections that follow.

Table 6 XMP Field Types

Field Type	Description
Simple	A simple field contains a single piece of information, eg a number, or a date
Structure	A structure is a container for one or more Simple fields or List
List	A list of values of the same type, eg: a list of numbers, or text strings. Lists may be defined as: • "Ordered" - the order in which the values are listed is important • "Unordered" – the order in which the values are listed in ***not*** important. • "Alternate" – you must chose one of the values listed. A special type of Alternate array is the "Language Alternate" array, which contains text in a number of different languages from which you can choose.

3.4.2 Simple XMP Fields

The field value can contain a variety of different types of information, and may have more than one value. For example, the value of the Reference Black and White field (nickname `xtif-ref-bw`) consists of six numbers. The types of data values that can be present in a field are shown in Table 7 "XMP Simple Data Types"[3].

Table 7 XMP Simple Data Types

Data Type	Description
Integer	A positive or negative whole number (eg 20).
Rational Number	A Positive or negative rational number, (eg 2.8)
Text String	A text string. For some fields the string may be any length up to a maximum of 2000 characters[4].
Choice	The field can contain one of a defined set of field values. eg the "White Balance" field can have a value of "Manual" or "Auto". The XMP Specification defined two types of "Choice" fields. *Open Choice* fields and *Closed Choice* fields. *Open Choice* fields can contain one of a set or predefined values, or another value entered by the user. *Closed Choice* fields can only contain one of the predefined values. EXIFutils does not currently support open choice fields, and treats them as Text fields in which any value can be entered.
Date/Time	A date/time in the format YYYY-MM-DDTHH:MM:SS-ZZ:ZZ. eg `2005-09-07T15:07:40-07:00`
Bool	The field can contain either "True" or "False"
Point	A set of two numbers separated by a comma, representing a point in a two dimensional array.
Rational Number List	A comma-separated list of rational numbers.

[3] The XMP Specification defines more data types. The EXIFutils provide a simplified view of these data types to the user.

[4] This limitation is imposed by the EXIFutils, not by the EXIF Specification. When running an unregistered version of EXIFutils, the maximum length to which a string can be set is 99.

3.4.3 List Fields

Lists are a list of values of the same type. For example the “Owner” field (nickname `xaux-owner`) contains a list of the names of the owners of the image (i.e. an list of text strings). There are three types of lists:

List Type	Description
Ordered	The order in which the list entries are stored is significant. For example, in the Camers Raw Tone Curve field (nickname `xraw-tone-curve`) is a list of points on a tone curve.
Unordered	The order in which list entries are stored is not significant. For example, the Rights Owner field (`rt-owner`) field consists of a list of text strings representing a list of the owners of the image. The order of the owner names in the list is not important and does not convey anything about the relationship between the owners.
Alternate	In an Alternate list, one of the values in the list is to be selected for display. See Section.4.2.2 “List Entries”. The most common form of Alternate list is the *Language Alternate* list. Fields defined as Language Alternate lists can contain text strings in multiple languages. One of the text strings is to be selected for display. The first entry in the list is the default value to be displayed if the user does not specify which language string they want to be displayed. See Section 4.2.3 “Selecting/Setting Entries in a Language Alternate List”.

3.4.4 Structure Fields

A Structure field is a field that consists of a group of subfields. Each subfield can be defined as any of the XMP field types described in the previous sections. EXIFutils does not currently support nested structure fields; i.e. structures as subfields of other structure fields is not supported.

A common structure field is the “Flash” field (nickname `xex-flash`) which contains a number of fields that describe the functioning of the flash. Subfields include Flash Fired (`fl-fired`) and Flash Mode (`fl-mode`).

See Section 4.2.1 “Structure Fields” for more information.

3.5 Syncronisation between XMP and non-XMP Data Fields

EXIFutils does not provide automatic syncronisation between XMP and non-XMP versions of EXIF and IPTC fields. Additional commands to synchronise fields are planned for a future release.

3.5.1 EXIF XMP Fields

The table below shows the relationship between EXIF fields and their XMP equivalents. Note that they values of some EXIF fields are split across multiple XMP fields and vice versa.

Table 8 Mapping of Legacy EXIF Fields to XMP

EXIF Field Nickname	XMP Field Nickname
aperture	xex-aperture
artist	xtif-artist
ave-comp	xex-comp-bit-pixel
brightness	xex-bright-value
cfa-pattern	xex-cfa-columns
	xex-cfa-rows
	xex-cfa-values
colour-space	xex-color-space
comment	xex-user-comment
comp-conf	xex-comp-config
contrast	xex-contrast
copyright	xtif-copyright
custom-render	xex-custom-render
date-digi	xex-date-digi
date-mod	xtif-date-time
date-taken	xex-date-taken
description	xtif-description
dev-setting	xex-dsd-columns
	xex-dsd-rows
	xex-dsd-settings
digital-zoom	xex-digital-zoom
exif-ver	xex-version
exp-bias	xex-exp-bias
exp-index	xex-exp-index
exp-mode	xex-exp-mode
exp-prog	xex-exp-prog
exp-time	xex-exp-time
file-src	xex-file-src
firm-ver	xtif-software
flash	xex-flash-fired
	xex-flash-ret
	xex-flash-mode
	xex-flash-func
	xex-flash-redeye
flash-energy	xex-flash-energy
flashpix-ver	xex-flashpix-ver

f-number	xex-f-number
focal-len	xex-focal-len
focal-len-35mm	xex-focal-len-35mm
fp-res-unit	xex-fp-res-unit
fp-x-res	xex-fp-x-res
fp-y-res	xex-fp-y-res
gain	xex-gain-control
gps-altitude	xex-gps-altitude
gps-alt-ref	xex-gps-alt-ref
gps-area	xex-gps-area-info
gps-dest-bear	xex-gps-dest-bear
gps-dest-bear-ref	xex-gps-dest-bear-ref
gps-dest-dist	xex-gps-dest-dist
gps-dest-dist-ref	xex-gps-dest-dist-ref
gps-dest-lat	xex-gps-dest-lat
gps-dest-long	xex-gps-dest-long
gps-diff-corr	xex-gps-diff
gps-geodetic	xex-gps-map-datum
gps-img-dir	xex-gps-img-dir
gps-img-dir-ref	xex-gps-img-dir-ref
gps-latitude	xex-gps-latitude
gps-longitude	xex-gps-longitude
gps-mode	xex-gps-measure-mode
gps-mov-dir	xex-gps-track
gps-mov-dir-ref	xex-gps-track-ref
gps-precision	xex-gps-precision
gps-proc-method	xex-gps-proc-method
gps-recv-speed	xex-gps-speed
gps-recv-stat	xex-gps-status
gps-satellite	xex-gps-satellites
gps-speed-unit	xex-gps-speed-ref
gps-time	xex-gps-timestamp
gps-ver	xex-gps-version
height	xex-pixel-y-dim
image-id	xex-unique-id
iso-speed	xex-iso-speed
light-src	xex-light-src
main-comp	xtif-compress
main-len	xtif-length
main-width	xtif-width
make	xtif-make
max-aper	xex-max-aper
meter-mode	xex-meter-mode
model	xtif-model
oecf	xex-oecf-columns
	xex-oecf-names
	xex-oecf-rows
	xex-oecf-values
orient	xtif-orient
photo-int	xtif-photo-inter

planar-conf	xtif-planar-conf
prim-chrom	xtif-primary-chrom
ref-bw	xtif-ref-bw
res-unit	xtif-res-unit
sample-pix	xtif-sample-per-pixel
saturation	xex-saturation
scene-mode	xex-scene-mode
scene-type	xex-scene-type
sense-method	xex-sense-method
sfr	xex-sfr-columns
	xex-sfr-names
	xex-sfr-rows
	xex-sfr-values
sharpness	xex-sharpness
shutter	xex-shutter
sound-file	xex-rel-sound-file
spec-sens	xex-spectral-sens
subj-area	xex-subj-area
subj-dist	xex-subj-dist
subj-locn	xex-subj-locn
subj-range	xex-sharpness
trans-func	xtif-trans-func
white-bal	xex-white-bal
white-point	xtif-white-point
width	xex-pixel-x-dim
x-res	xtif-x-res
ycbcr-coeff	xtif-ycbcr-coeff
ycbcr-posn	xtif-ycbcr-posn
ycbcr-sub-samp	xtif-ycbcr-sub
y-res	xtif-y-res

3.5.2 IPTC XMP Fields

The mapping between legacy IIMv4 IPTC fields and their XMP equivalents is described in “IPTC Standard Photo Metadata 2008, IPTC Core Specification Version 1.1, IPTC Extension Specification Version 1.0, Document Revision 2”.

Legacy IPTC fields are mapped to fields from several different XMP schemas. Table 9 shows these mappings and the EXIFutils nicknames for both legacy and XMP fields.

Table 9 Mapping of Legacy IPTC Fields to XMP

Traditional IPTC IIMv4		XMP IPTC Core		Photoshop CS File Info… Name
IPTC Field Name	**EXIFutils Nickname**	**Schema**	**EXIFutils Nickname**	
Urgency	`ip-urgency`	Photoshop	`xph-urgency`	Urgency
Category	`ip-category`	Photoshop	`xph-category`	Categories
Supplemental Categories	`ip-suppcat`	Photoshop	`xph-supp-cat`	Suppl. Categories
Keywords	`ip-keyword`	Dublin Core	`xdc-subject`	Keywords
Special instructions	`ip-instructions`	Photoshop	`xph-instructions`	Instructions
Date Created	`ip-date`	Photoshop	`xph-date-created`	Date Created
By Line	`ip-byline`	Dublin Core	`xdc-creator`	Author
By-Line Title	`ip-byline-title`	Photoshop	`xph-author-posn`	Authors Position
City	`ip-city`	Photoshop	`xph-city`	City
Country/Primary location name	`ip-country`	Photoshop	`xph-country`	Country
Original Transmission Reference	`ip-orig-ref`	Photoshop	`xph-trans-ref`	Transmission Reference
Headline	`ip-headline`	Photoshop	`xph-headline`	Headline
Credit	`ip-credit`	Photoshop	`xph-credit`	Credit
Source	`ip-source`	Photoshop	`xph-source`	Source
Copyright Notice	`ip-copyright`	Dublin Core	`xdc-rights`	Copyright Notice
Caption/Abstract	`ip-caption`	Dublin Core	`xdc-description`	Description
Writer/Editor	`ip-cap-writer`	Photoshop	`xph-caption-writer`	Description Writer
New fields, not present in legacy IPTC data.		Iptc4XmpCore	`i4x-cr-city`	
		Iptc4XmpCore	`i4x-cr-country`	
		Iptc4XmpCore	`i4x-cr-address`	
		Iptc4XmpCore	`i4x-cr-postal-code`	
		Iptc4XmpCore	`i4x-cr-state-prov`	
		Iptc4XmpCore	`i4x-cr-email`	
		Iptc4XmpCore	`i4x-cr-phone`	
		Iptc4xmpCore	`i4x-country-code`	
		Iptc4XmpCore	`i4x-genre`	
		Iptc4XmpCore	`i4x-location`	
		Rights	`xrt-usage-terms`	
		Iptc4xmpCore	`i4x-subj-code`	
		Iptc4XmpCore	`i4x-scene`	
Edit Status	`ip-edit-status`	Deprecated fields. Not supported in XMP data.		
Fixture Identifier	`ip-fixture`			
Content Location Name	`ip-cont-loc-name`			
Contact	`ip-contact`			

4 Referencing Fields Using Nicknames

4.1 Metadata Nicknames

Several of the EXIFutils commands allow you to specify a list of fields to display, edit, copy, or delete. To provide an easy way of referring to EXIF fields, the EXIFutils assign a *nickname* to each metadata field. A complete list of field nicknames is provided in EXIFutils Field Reference Guide.

4.2 Nicknames for XMP Structure and List Fields

4.2.1 Structure Fields

EXIFutils assigns a nickname to the whole structure field and to the individual fields within the structure. For example, the Flash structure fields includes a number of subfields

- Flash structure (nickname `xex-flash`)
    - Fired (`fl-fired`)
    - Returned (`fl-ret`)
    - Mode (`fl-mode`)
    - Function (`fl_func`)
    - RedEyeMode (`fl-redeye`)

To access fields within the structure, you must precede the field name with the name of the structure and a dot '.'. For example to access the `fl-fired` field you enter `xex-flash.fl-fired`. The commands below demonstrate this use:

```
exiflist -o ln -f xex-flash.fl-fired myfile.jpg
exifedit -a xex-flash.fl-fired=true myfile.jpg
```

To list all fields in a structure you can specify the structure name with no subfields. For example:

```
exiflist -o ln -f xex-flash myfile.jpg
xex-flash.fl-fired="False",xex-flash.fl-return="No Strobe Return Detection",xex-
flash.fl-mode="Compulsory Flash Suppression",xex-flash.fl-function="False",xex-f
lash.fl-redeye="False"
```

4.2.1.1 Adding and Deleting Structure Fields

When adding a structure subfield it is not necessary to explicitly add the structure field. If the structure field does not exist, then EXIFutils will create it when you add the first subfield in the structure. For example, if the Flash structure field (nickname `xex-flash`) is not present and you add the Flash Fired subfield (nickname `fl-fired`), then EXIFutils will automaticall create the Flash structure field and insert the Flash Fired field into it.

Likewise when you delete the last subfield in a structure, EXIFutils will automatically delete the structure field.

It is possible delete a complete structure field and all of its subfields by deleting the structure field by name. For example, to delete all of the fields in the Flash structure field, delete the structure field (`xex-flash`).

4.2.2 List Entries

One nickname is used to reference all entries in a list field. To distinguish the list entry is to be retrieved, added, or deleted square a reference value is appended to the nickname surrounding by square brackets "`[ ]`". The reference value is either an integer number zero or greater (eg `xex-iso-speed[1]`), or in the case of Language Alternate lists a language specification (eg `xdc-description[en]`). When numeric references are used the first entry in the list is entry 1 (i.e. `[1]`). The meaning of numeric reference values is described in Table 10 "Referencing XMP List Field Entries".

Table 10 Referencing XMP List Field Entries

Reference Value	Example	Meaning When…		
		Retrieving List Value (exiflist -f option)	**Adding List Value (exifedit -a option)**	**Deleting List Value (exifedit -r option)**
none	`xdc-description`	If the list is an “Alternate” list then the first entry is retrieved, otherwise all list values are retrieved.	Add value to the end of the list	Delete all list entries
`[]`	`xdc-description[]`			
`[*]`	`xdc-description[*]`			
`[0]`	`xdc-description[0]`	Invalid. Returns null value.	Insert the entry at the *beginning* on the list.	Invalid. Error “Invalid array subscript” will be reported.

`[n]`	`xdc-description[1]`
	`xdc-description[2]`
	`xdc-description[99]`
		Retrieves the nth value in the list. The first entry in the list is [1].
		If *n* is higher than the number of entries in the list then an empty string is returned.
			Overwrites the nth value in the list.
			If *‘n’* is high than the number of entries in the list the the value will be added to the end of the list.
				Deletes the nth value in the list. Values later in the list will be moved forward.
				Eg: If you delete the 4th entry in the list then the 5th entry becomes the 4th entry, and so on.
				If ‘*n*’ is higher than the number of entries in the list then the last entry will be deleted.

4.2.3 Selecting/Setting Entries in a Language Alternate List

Language Alternate fields allow you to store versions of a text value in different languages. The version of the text to be displayed, added or deleted is selected by using a language/country code as the reference value.

The first entry added to the list is the default language value for this field. This is the value that is retrieved if no language is specified. When the first entry is added to a Language Alternate list, two entries will actually be added:

1. The first entry will be added with a language value set to ‘`x-default`’.
2. A second entry will be added with the same text value as the default entry and the language set to the language you specified (or the default language “en” if no language was specified).

For example, if the Language Alternate field `xdc-description` is not currently present, then the following command

```
exifedit -a “xdc-description[en]=Hello” myfile.jpg
```

will result in the following entries being added to the `xdc-description` field:

```
    description[1]          :   Hello
      lang                  :     x-default
    description[2]          :   Hello
      lang                  :     en
```

Table 11 describes the use of language references to display, add, and delete language alternate values.

It is also possible to access Language Alternate list entries using the numeric reference values described in Table 10 “Referencing XMP List Field Entries”.

Table 11 Referencing Language Alternate Fields

Reference Value	Example	Meaning When…		
		Retrieving List Value (exiflist -f option)	**Adding List Value (exifedit -a option)**	**Deleting List Value (exifedit -r option)**
none	`xdc-description`	Retrieve the default language value (the first entry in the list)	Adds an entry in the default language to the end of the list. If this is the first entry in the list then two entries are added: one with language ‘`x-default`’ followed by an entry with language set to the EXIFutils default language `[en]`. See Section 4.2.3.2 for how to change the EXIFutils default language to a language other than English.	Delete all list entries.
`[]`	`xdc-description[]`			
`[*]`	`xdc-description[*]`	Retrieve all list entries		
[*lang*]	`xdc-description[en]`	Retrieve the entry that matches the specified language code. If no entry matches the language code, the the default (first) entry will be returned.	If there is an existing entry for the specified language then it will be replaced. If the existing entry is the default (first) entry, then the default entry will also be replaced. If there is no existing entry then a new entry will be added to the end of the list.	Invalid. Deleting of individual language entries not yet supported.
[*lang*] [*lang-country*]	`xdc-description[en-AU]`	Retrieve the entry that matches the specified language/country code. If no entry matches the exact language/country, but there is another entry that matches the language code, then it will be returned. If there are no entries that match the language code, then the default (first) entry will be returned.	If there is an existing entry for the specified language/country then it will be replaced. If the existing entry is the default (first) entry, then the default entry will also be replaced. If there is no existing entry then a new entry will be added to the end of the list. Other entries for the same language (eg ‘en’, ‘en-US’) will be left unchanged.	

4.2.3.1 Language/Country Codes

This section provides a simplified description of the format of the language country codes used to reference Language Alternate fields. For a full description refer to RFC3066 (see http://www.ietf.org/rfc/rfc3066.txt).

In general, a language reference consists of either:

- a two letter language code (eg ‘en’)
- a two letter language code followed by a hypen and a two letter country code.

The convention is that language codes are lower case and the country codes are upper case, although the case does not affect the processing of the language reference. The following are examples of value language references:

- ‘en’ (English)
- ‘en-US’ (English – United States)
- ‘en-CA’ (English – Canada)
- ‘fr’ (French)

4.2.3.2 EXIFutils Default Language

If no language reference is specified when adding a Language Alternate entry, EXIFutils assumes that the language is English (i.e. the language references is assumed to be `[en]`).

The default language can be changed by setting the `EXIFUTILS_XMP_LANG` environment variable to your required default language. For example, to set the default language to French, you would set the `EXIFUTILS_XMP_LANG` to ‘`fr`’. To see the value to which the default language is current set use the `exifinfo` command to display global settings:

```
exifinfo -g
```

4.2.3.3 XMP Language Alternate List Usage Examples

The following examples demonstrate the setting, removal, and displaying of Languge Alternate values.

Example 1

Remove the `xdc-description` and all the list entries in it

```
exifedit -r xdc-description myfile.jpg
```

Add value with no language reference. This will default to language `[en]` and creates an `[x-default]` value with the same value

```
exifedit -a xdc-description=Hello myfile.jpg
```

Display the value

```
exiflist -f xmp myfile.jpg

[...]
XMP Dublin Core
  description              : [Sorted List of Alternative Values]
    description[1]         :   Hello
      lang                 :     x-default
    description[2]         :   Hello
      lang                 :     en
```

Example 2

Remove the `xdc-description` and all the list entries in it

```
exifedit -r xdc-description myfile.jpg
```

Add value with an Australian English language reference. As this is the first entry to be added it will be the default language and an `x-default` language entry will be created with this value.

```
exifedit -a xdc-description[en-AU]=Gidday myfile.jpg
```

Add a generic English language value (language reference `[en]` with no country variant specified). This will NOT be the default language value as it was not the first value added.

```
exifedit -a xdc-description[en]=Hello myfile.jpg
```

Add a French language value (language reference `[fr]` with no country variant specified). This will NOT be the default language value as it was not the first value added.

```
exifedit -a xdc-description[fr]=Bonjour myfile.jpg
```

Display all list values

```
exiflist -f xmp myfile.jpg

[...]
XMP Dublin Core
    description[1]         :   Gidday
      lang                 :     x-default
    description[2]         :   Gidday
      lang                 :     en-AU
    description[3]         :   Hello
      lang                 :     en
    description[4]         :   Bonjour
      lang                 :     fr
```

Display the `xdc-description` field with no language specified. This will retrieve the default language value

```
exiflist –o ln -f xdc-description myfile.jpg
xdc-description="Gidday"
```

Display the `xdc-description` field for the French language. This will retrieve the French language value.

```
exiflist –o ln -f xdc-description[fr] myfile.jpg
xdc-description="Bonjour"
```

Display the `xdc-description` field for the French Canadian language (language reference `[fr-CA]`. As these is no specific French Canadian entry, the closest language (generic French) will be retrieved.

```
exiflist –o ln -f xdc-description[fr-CA] myfile.jpg
xdc-description="Bonjour"
```

Display the `xdc-description` field referencing a language (Italian) for which, in this example, there is no entry in the list. This will retrieve the default language value (Australian English).

```
exiflist –o ln -f xdc-description[it] myfile.jpg
xdc-description="Gidday"
```

Example 3

Prior to entering these commands the `EXIFUTILS_XMP_LANG` environment variable was set to ‘`fr`’. This causes any values that are added without specifying a language reference will by default use the `[fr]` language reference. The method of setting this value depends on the operating system on which you are running.

Remove the `xdc-description` and all the list entries in it

```
exifedit -r xdc-description myfile.jpg
```

Add value with with no language reference. As the EXIFutils default language has been set to French, this entry will be added with a language reference of `[fr]`. As this is the first language entry to be added to this field an `x-default` language entry will also be created with this value.

```
exifedit -a xdc-description=Bonjour myfile.jpg
```

Display all list values

```
exiflist -f xmp myfile.jpg

[...]
XMP Dublin Core
    description[1]          :   Bonjour
      lang                  :     x-default
    description[2]          :   Bonjour
      lang                  :     fr
```

4.3 Selecting EXIF Fields Using “Tag” Value

EXIFutils allocates a nickname for all EXIF fields that known at the time of release. To access fields for which no nickname has been allocated EXIFutils provides a means of retrieving a field using its 4 digit hexadecimal tag number. This is done by specifying the field name as ‘`tag-`*`xxxx`*’, where *`xxxx`* is the hexadecimal tag number.

For example, if the tag number is `00af`, then the field name would be `tag-00af`. Below is an example showing this field used in an `exiflist` command:

```
exiflist -o l -f file-name,tag-00af myfile.jpg
```

If there is more than one field with the same tag in a file, then the first occurrence of the tag will be retrieved.

4.4 Nicknames for Derived File Attributes

In addition to the fields defined by the supported metadata specifications, EXIFutils provides additional nicknames that represent information that is derived from the file attributes or other meta properties. There include:

- File attribute information (see Table 12 File Attribute Nicknames)
- Individual parts of the EXIF Date Taken field (see Table 13 EXIF Date Taken Subfield Nicknames)
- The names of files created by the EXIFutils (see Table 14 Created File Nicknames)

Table 12 File Attribute Nicknames

Nickname	Description
`file-name`	The name of the image file *excluding* the directory path, eg `myphoto.jpg`
`file-name-full`	The name of the image file *including* the directory path, eg `/HolidayPhotos/myphoto.jpg`
`file-base`	The part of the image file name that precedes the last dot '.'. For example, if the `file-name` is '`myphoto.jpg'`, then `file-base` will contain '`myphoto`'.
`file-ext`	The file name extension, i.e. the part of the file name that follows the last dot '.'. For example, if the `file-name` is '`myphoto.jpg'`, then `file-ext` will contain '`jpg`'.
`file-dir`	The directory in which the image file is located, eg `/HolidayPhotos`
`file-size`	The size of the file in bytes
`file-date-mod`	The date/time that the file was last modified. The date is in EXIF standard format, i.e. "YYYY:MM:DD HH:MM:SS"
`file-date-created`[5]	The date the file was created. The date is in EXIF standard format, i.e. "YYYY:MM:DD HH:MM:SS"

Table 13 EXIF Date Taken Subfield Nicknames

Nickname	Description
`yyyy`	Year part of `date-taken` field. Four digits. eg: "2005"
`yy`	Year part of `date-taken` field. Two digits. eg: "05"
`mm`	Month part of `date-taken` field. Two digits. eg: "12"
`mmm`	Month part of `date-taken` field. Three letters. eg: "dec"
`dd`	Day part of `date-taken` field. Two digits. eg: "31"
`hr`	Hour part of `date-taken` field. Two digits, using 24 hour clock. eg: "23"
`min`	Minute part of `date-taken` field. Two digits. eg: "59"
`sec`	Seconds part of `date-taken` field. Two digits. eg: "45"

NOTE: If the `date-taken` field is not present in a file, then the above nicknames will return an empty string.

[5] `file-date-created` is only available on Windows systems. On MacOS X, and Linux systems this field will contain the date of last state change of the file.

Table 14 Created File Nicknames

Nickname	Description
`file-name-audio`	The default name of the audio file that will be created by the `exiflist -w` command
`file-name-backup`	The name of the backup file that will be created by the `exifcopy` and `exifedit` commands.
`file-name-exi`	The default name of the EXIF Save file that will be created by the `exiflist -e` command
`file-name-thumb`	The default name of the thumbnail file that will be created by the `exiflist -t` command

4.5 Selecting Multiple Fields

In addition to specifying individual field names, the `exiflist` command provides two quick ways to select multiple meta data fields:

- **Wildcards:** a wildcard character can be used to select all fields whose nickname start with a certain string. For example '`gps-*`' will select all GPS related fields because the nicknames of all GPS related fields start with '`gps-`'. The wildcard character '*' can only appear at the end of the string.
- **Special Nicknames:** There are six special nicknames that can be used to select commonly used groups of fields:
  - `all:` Selects all supported metadata fields present in the image file. Refer to EXIFutils Field Reference Guide for a list of supported fields.
  - `exif:` Selects all non-XMP EXIF fields present in the image file. Refer to EXIFutils Field Reference Guide for a complete list of EXIF fields[6].
  - `exif-common:` '`exif-common`' is predefined to select the most commonly used EXIF fields.
  - `iptc:` Selects all IPTC fields. Refer to EXIFutils Field Reference Guide for a complete list of IPTC fields[7].
  - `rmeta:` Selects all Ricoh RMETA fields. Refer to EXIFutils Field Reference Guide for a complete list of RMETA fields.
  - `xmp:` Selects all meta data fields stored in XMP format. Refer to EXIFutils Field Reference Guide for a complete list of XMP fields.

[6] The `exif` nickname does ***not*** select EXIF fields stored in XMP format. EXIF fields stored in XMP format have a separate nickname.

[7] The `iptc` nickname does ***not*** select IPTC fields stored in XMP format. IPTC fields stored in XMP format have a separate nickname.

5 Working with Dates and Times

EXIF, IPTC, and XMP date/time fields are stored in a different formats.

All EXIF date/time fields include both the date and the time, in the following format:

> “`YYYY:MM:DD HH:MM:SS`”, eg: “`2004:01:27 21:53:00`”

You cannot omit the time.

IPTC dates and times are stored in separate fields, eg the `ip-date` and `ip-time` fields.
IPTC date fields are stored in the following format: “`YYYYMMDD`”, eg: “`20040127`”
IPTC time fields are stored in the following format: “`HHMMSS+/-ZZZZ`”, eg: “`212300+10000`”

where `HHMMSS` is the time in the local timezone, and `+/- ZZZZ` indicates the timezone. For example, “`090000-0500`” represents 9:00am in New York (5 hours ahead of GMT/UTC).

XMP date/time fields contain both the date and the time in the following format:

> “`YYYY-MM-DDTHH:MM:SS-ZZ:ZZ` “, eg:“2006-04-10T13:37:10-07:00”

You can use the `exifedit` command to set the value of date, time, and date/time fields. When you are setting the value of a date, time, or date/time field, `exifedit` does not require that the date and/or time be entered in the format in which it will be stored. `exifedit` accepts a number of date/time formats and converts them to the format required by the field you are setting.

The sections that follow describe the date/time formats that are accepted by `exifedit`.

5.1 Dates Formats

Table 15 shows the date formats that are accepted by `exifedit`.

Table 15 Accepted Date Formats

YYYY-MMM-DD
DD-MMM-YYYY
DD-MM-YYYY
YYYY-MM-DD
DDMMYYYY
YYYYMMDD
YY-MM-DD
DD-MM-YY
YYMMDD
DDMMYY

where:

- `YYYY` is the year, eg: `2004`
- `YY` is the year within the century, eg: “`04`”
- `MM` is the month number, eg: `05` for May
- `MMM` is the three-character month abbreviation. Valid values are “Jan”, “Feb”, “Mar”, “Apr”, “May”, “Jun”, “Jul”, “Aug”, “Sep”, “Oct”, “Nov”, and “Dec”. Values are not case sensitive; for example “FEB” and “feb” are valid months.
- `DD` is the day of the month, eg: `01`

Where a dash separator is shown in Table 15, any of the following separators will be accepted: “`:`”, “`.`”, or “`/`”. For example “`01:04:2004`” and “`01/04/2004`” are valid dates.

5.2 Time Formats

Table 16 shows the time formats that are accepted by `exifedit`.

Table 16 Accepted Time Formats

HH-MM-SS
HHMMSS
HH-MM
HHMM
HH-MM-SS±ZZZZ
HHMMSS±ZZZZ
HH-MM±ZZZZ
HHMM±ZZZZ
HH-MM-SS ±ZZZZ
HHMMSS ±ZZZZ
HH-MM ±ZZZZ
HHMM ±ZZZZ

where:

- HH is the hour in 24 hour notation, eg: “13” means 1pm
- MM is the minutes within the hour, eg: “56”
- SS is the seconds within the minute, eg: “12”
- ±ZZZZ is the timezone, eg: “+0500”, “-1000”.

Where a dash separator is shown in Table 16, the following separators will be accepted: “`:`”, “`.`”, or “`/`”. For example “`12:04:00`”, “`12.04.00`” and “`12/04/00`” are valid times.

When no SS (seconds) value is specified, an SS value of “00” will be assumed. Where no timezone is specified a time value of “+0000” will be assumed.

5.3 Ambiguous or Incomplete Dates and Times

If an incomplete or ambiguous date or time value is specified, the following rules will be used to interpret the value:

- If a date/time field is being set and only a date value has been supplied, the time part of the field will be set to midnight (00:00:00). If you only supply a time for a date/time field, an error will be reported and the field not set.
- If no timezone is specified for a field that requires a timezone, then GMT is assumed (i.e. +0000)
- If the ‘seconds’ part of a time value is omitted then a ‘seconds’ value of ‘00’ is assumed.
- If a two-digit year is specified (eg: “04, “99”) `exifedit` will attempt to select the correct century. If the two-digit year is less than “20” the date is assumed to be in the 21st century, eg. “04” is assumed to mean “2004”. If the two-digit year is “20” or geater it is assumed to be in the 20th century, eg: “99” is assumed to be “1999”.

Note that in some cases it will not be possible for EXIFutils to unambiguously determine your intended value. For exampled, the value “23:59:59” is unambiguously a time value, but “04:10:01 can either be October 1st 2004, or a time in the early morning (04:10:01AM). In cases where the specified value is both a valid date and a valid time, EXIFutils will look at the type of field being set to determine what was intended. If the field being set is a date field or a date/time field, the value will be interpreted as a date. If the field being set is a time then the value will be interpreted as a time.

EXIFutils will also attempt to interpret dates in both the American ‘month first’ format (eg MM:DD:YY) and the international format (DD:MM:YY), however it is not always possible to determine which is intended. For example, 03:01:04 could mean either March 1st 2004 or January 3rd 2004. In these cases EXIFutils assumes the international format as it most closely matches the date formats used by EXIF and IPTC fields.

5.4 *Converting Between EXIF, XMP, and IPTC Date/Time Formats*

It is sometimes necessary to copy EXIF or XMP date/time field values into IPTC date and time fields, and vice versa. EXIF and XMP date fields contain both the date and time, whereas IPTC fields hve separate fields for dates and times. To simplify this task, EXIFutils has been designed to automatically convert between EXIF, XMP, and IPTC date time formats.

5.4.1 Setting IPTC Date and Time fields to the value of EXIF and XMP date/time fields

An IPTC date field (eg: the `ip-date` field) can be set to the date part of the EXIF `date-taken or XMP xex-date-taken` field by simply assigning the EXIF or XMP field value to the IPTC field, for example:

```
exifedit -a “ip-date={date-taken}” myfile.jpg
exifedit -a “ip-date={xex-date-taken}” myfile.jpg
```

`exifedit` will automatically extract the date part of the `date-taken or xex-date-taken` field and convert it to the correct format for the `ip-date` field.

Similarly, an IPTC time field (eg `ip-time`) can be set to the time part of the EXIF `date-taken` or XMP `xex-date-taken` field, for example:

```
exifedit -a “ip-time={date-taken}” myfile.jpg
exifedit -a “ip-time={xex-date-taken}” myfile.jpg
```

`exifedit` will extract the time part of the `date-taken` or `xex-date-taken` field and convert it to the IPTC time format. Note that EXIF date/time fields do not contain any time zone information, unlike IPTC time fields. `exifedit` allows you to add time zone information when setting an IPTC time field, for example:

```
exifedit -a “ip-time={date-taken} +1000” myfile.jpg
```

5.4.2 Setting EXIF and XMP date/time fields to the value of IPTC date and time fields

EXIF and XMP date fields (eg `date-taken`, `xex-date-taken`) contain both a date and a time portion. IPTC Meta data contains dates and times in separate fields (eg: `ip-date`, `ip-time`). `exifedit` allows you assign EXIF and XMP date fields to the combined value of IPTC date and time fields. For example, to set the EXIF `date-taken` field to the date and time specified in the `ip-date` and `ip-time` fields, use the following command:

```
exifedit -a “date-taken={ip-date} {ip-time}” myfile.jpg
```

`exifedit` automatically converts the IPTC date and time fields into the format required by the EXIF `date-taken` field.

6 Character Sets

6.1 Introduction

A “character set” is an encoding scheme that defines how text characters are stored as binary values. There are many different “standard” character sets that can be used to store text data in files.

The EXIF Specification states that EXIF text fields are to be stored using the “7 bit ASCII” character set, which means that only unaccented English characters and punctuation marks can be used[8]. The IPTC Standard supports a wide variety of character sets, but the mechanism is limited.

As a result many image editing applications ignore the character sets required by the EXIF and IPTC Standards. Instead they store text data using extended character sets that allow the use of a wide range of non-English characters. It is not always possible to determine which character set was used to store the EXIF and IPTC data in an image file.

The character set most widely used for EXIF and IPTC fields is the Latin1 character set (ISO8859-1) which allows the representation of more than 40 Western European languages. EXIFutils therefore by default assumes that EXIF and IPTC fields are stored using Latin1, but provides the option to use different character sets if required.

The sections that follow describe how EXIFutils handles characters sets, and how the user can specify the character set to be used.

6.2 How EXIFutils Handles Character Sets

EXIFutils commands convert all character strings extracted from both EXIF and IPTC fields into a common character set for internal processing and for display. In EXIFutils terminology, this is called the ***Command*** character set. This is the character set that will be used when outputing text to the command terminal window or a file. It is also the character set that is used to interpret text files that are read by EXIFutils (eg when a template file is passed as a parameter to `exiflist`). The command character set must therefore match the character set that that the underlying operating system expects.

To correctly read, set, and display text strings, EXIFutils needs to know the character set to use for EXIF fields, IPTC fields, and the Command Character set. The default character sets used by EXIFutils are shown in Table 17. These defaults can be overridden using in the following ways:

- using the `-p` option on the command line
- setting EXIFutils environment variables

Table 17 Default Character Sets

Platform	EXIF Character Set	IPTC Character Set	Command Character Set
Microsoft Windows	LATIN1	LATIN1	ANSI Code Page[9] (except when `exiflist` is writing directly to the Command Prompt window, in which case the OEM Code Page is used).
Mac OSX			LATIN1 (ISO8859-1)
Linux			

[8] The EXIF comment field is an exception to this rule. It can be stored in either ASCII or UNICODE. Refer to Section 6.3 “EXIF Comment Field” for more information.

[9] On Windows, character sets are translated by Code Pages. There are two special code pages defined in the system; the default ANSI Code Page, and the OEM Code Page.

6.2.1 Using the `-p` Option to Specify Character Sets

The `exifcopy`, `exifedit`, `exiffile`, and `exiflist` commands all accept the `-p` option to specify the character sets to be used. The `-p` has the following form:

```
-p cmd=charset,exif=charset,iptc=charset
```

where *`charset`* is the name of a valid character set . Any of the three character set types (`cmd`, `exif`, or `iptc`) may be omitted, in which case they will take the default value, or the values specified via environment variables (see 6.2.2 “Using Environment Variables to Specify Character Sets”). Note that on the `exifedit` command, the `-p` options ***must*** be before the `-a` (add fields) option.

Table 18 shows commonly used character set names, including special values defined by EXIFutils. EXIFutils will accept any character set name listed by the standard `iconv -l` command, although not all character sets will produce meaningful results. The Command (cmd=) character set cannot be set to UNKNOWN.

Table 18 Valid Character Set Names

Character Set Name	Comments
LATIN-1 or LATIN1	Code Page 28591 on Windows
UTF-7 or UTF7	
UTF-8 or UTF8	
MAC	Mac Roman character set. Commonly used on Mac OSX systems.
OEM	Default OEM Code Page (Windows only)
ACP	Default ANSI Code Page (Windows only)
number	The number of any Code Page that is present on the system (Windows only)
Special Values Defined by EXIFutils	
UNKNOWN	Character set is not known. EXIFutils will not perform any translation. Not valid as a Command Character set (i.e. `cmd=unknown` is not valid).
LOCALE	The default character set for the current locale (OSX, Linux, Solaris only)

The following are examples of valid `-p` option settings:

```
-p exif=utf8,iptc=mac
-p
cmd=locale,exif=utf7,iptc=unknown
-p exif=latin1
-p iptc=unknown
```

While EXIFutils will accept any valid character set name only characters sets which meet the following criteria will function correctly:

- the lower order characters (0x00 to 0x7f) are the standard ANSI characters,
- there are no embedded null byes (i.e. null can be used to represent the end of the string).

6.2.2 Using Environment Variables to Specify Character Sets

If the same character sets are to be used repeatedly, environment variables can be used to permanently set the character set so that the `-p` option does not need to be included on every command.

Table 19 shows the environment variables used to set EXIFutils character sets. If an environment variable and the `-p` option are used, then the `-p` option overrides the environment variable.

Table 19 Character Set Environment Variables

Environment Variable	Meaning
`EXIFUTILS_CS_CMD=`*`charset`*	Equivalent to `-p cmd=`*`charset`*
`EXIFUTILS_CS_EXIF=`*`charset`*	Equivalent to `-p exif=`*`charset`*
`EXIFUTILS_CS_IPTC=`*`charset`*	Equivalent to `-p iptc=`*`charset`*

If both the `-p` option and the equivalent environment variable are specified, then the `-p` option setting is used.

6.2.3 Windows Batch (.BAT) Files and the -y Option

Problems can arise when using non-English characters in EXIFutils command line parameters within Windows .BAT files. Most text editors create text files (include .BAT files) using the ANSI Code Page, however Windows expects .BAT file to be created using the OEM Code Page. This means that some characters may not be interpretted correctly. If you find that a command works when entered in the Command Line windows, but does not work within a .BAT file then this is likely to be the cause.

The solution is to add the `-y` option as the first option on EXIFutils commands within batch files. The `-y` option tells the EXIFutils commands to translate the command line options so that they are interpretted correctly. The `-y` option is valid on the `exifedit`, `exiffile`, and `exiflist` commands.

6.2.4 Troubleshooting Character Set Problems

If EXIF or IPTC text values are not being displayed correctly, then it is likely that one or more characters sets have not been set correctly. The most likely problems is that the character set used to store the EXIF or IPTC value is different to the EXIFutils default character set (LATIN1). The most common character sets other than LATIN1 used in storing text values are UTF8 and MAC, the later being most likely if the file was created on a MacOS X System.

To determine if this is the case, try changing the character set to one of these other values. For example, if IPTC text fields are not display correctly, try one of the following:

```
exiflist -p iptc=mac myfile.jpg
exiflist -p iptc=utf8 myfile.jpg
```

If neither of these work, contact EXIFutils support for assistance (support@hugsan.com), sending a sample of the image file.

6.3 EXIF Comment Field

The EXIF comment field (nickname `comment`) can be stored using either plain ASCII text (which only allows Western European characters), or UNICODE (which supports a wide range of international characters)[10]. By default the comment field will be stored using ASCII.

To store the `comment` field using UNICODE, use the `-u` option on the `exifedit` command. For example:

```
exifedit -u -a comment=hello myfile.jpg
```

Alternatively, if you want exifedit to always use UNICODE for the comment field, set the `EXIFUTILS_UNICODE_COMMENT` environment variable to 'Y'. Refer to the documentation for you operating system for details of who to set environment variables.

Note that the character set used to store the EXIF comment field is not affected by the `-p exif=charset`, it is only affected by the `-u` option and the `EXIFUTILS_UNICODE_COMMENT` environment variable.

[10] The EXIF Specification also allows use of the JIS Japanese character set, but this is not supported by EXIFutils.

7 Templates

Templates are a powerfully feature that allows you to substitute field values (EXIF, IPTC, RMETA), and file attribute values into a character string. Three EXIFutils commands allow you to specify a template in place of a fixed character string (Table 20).

Table 20 Command Options That Accept Templates

Command	Option	Option Description	Use of Template
`exifedit`	`-a`	Add EXIF field to image	The value to which a field is set can be specified as a template.
`exiffile`	`-n`	Rename file	The name to which an image file is renamed can be specified as a template.
`exiflist`	`-s`	Set output template	The template defines the output format used to display fields.
	`-t`	Extract thumbnail	The template specifies the name of the file into which the thumbnail is extracted.
	`-w`	Extract audio stream	The template specifies the name of the file into which the audio stream is extracted.
	`-e`	Extract Meta data	The template specifies the name of the file into which the Meta data is extracted.

In its simplest form a *template* is a character string into which values are substituted. A template character string can contain:

1. markers that indicate where field values are to be inserted
2. formatting characters
3. other text

A *marker* is any valid EXIFutils field nickname surrounded by curly brackets, eg “`{f-number}`”. See Section EXIFutils Field Reference Guide for a complete list of nicknames.

Formatting characters are special characters preceded by a back-slash character “\”. Valid formatting characters are shown in Table 21.

Table 21 Template Formatting Characters

Character	Meaning
`\n`	New line character
`\r`	Carriage Return[11]
`\t`	Tab character
`\\`	Backslash character

The EXIFutils commands evaluate a template by scanning for markers and replacing them with the values of the corresponding fields. Any quote characters within a substituted field value will be preceded by a backslash escape character “\”. If the field name specified in the marker is a valid field name, but that field is not present in the image file, then a empty string is substituted. Formatting characters are replaced with the corresponding format character. Any sequence of characters surrounded by curly brackets that is not a recognized nickname will be left unchanged.

[11] Carriage return characters are used to indicate a new line on MacOS X systems.

Example 1

The `exifedit` command below sets the value of the `comment` field based on the value of the `shutter` speed and `f-number` fields:

```
exifedit -a “comment=Shutter: {shutter}\tF-stop: {f-number}\n” photo.jpg
```

If the value of the `shutter` field is “`1/125`”, and the value of the `f-number` field is “`5.6`”, then when the above template is evaluated, the `comment` field will be set to the following string:

```
Shutter: 1/125,   F-stop: 5.6
```

Example 2

The `exiffile` command below renames an image file so that the new file name contains the camera model followed by the original file name:

```
exiffile -n {model}-{file-name} photo.jpg
```

If the value of the `model` field is “`NIKON D1X`”, then the resulting file name will be:

```
NIKON D1X-photo.jpg
```

7.1 Sectioned Templates

To provide greater control over the output format when displaying fields from multiple files, the `exiflist -s` option accepts a more advanced form of template, called a *sectioned template*. A sectioned template has all the features of a basic template but in addition can be divided into three sections:

- The *head* section, which contains text that is to be displayed once, before fields from any file are displayed
- The *file* section, which contains text that is to be displayed once for each file
- The *tail* section, which contains text that is to be displayed once, after all files have been displayed

These sections allow header and trailer information to be displayed before and after the details of the files being displayed.

Section markers in the template indicate the beginning of each section:

```
{EXIF-HEAD}

lines to be display once, before the first image file is displayed

{EXIF-FILE}

lines to be displayed once for each image file that is displayed

{EXIF-TAIL}

lines to be display once, after the last image file has been displayed
```

Section markers

Any section can be omitted if not required. If no section markers are included, then all lines in the template are considered to be in the *file* section. The section markers are not shown in the resulting output.

Example 3

The commands and templates below produce an HTML page that contains a table with headings on each column, and one row for each image file in a directory. Each row will contain the image file name as a web link, the time the image was taken, its file size, and image resolution. The following `exiflist` command and template will achieve this:

```
exiflist -o t -s @template.txt mydir > photolist.html
```

`template.txt` contains:

```
{EXIF-HEAD}
<HTML>
<HEAD>My Photographs</HEAD>
<TABLE ID="Table1" BORDER=1 CELLSPACING=3 CELLPADDING=1 WIDTH=725>
<TR>
  <TD>File Name</TD>
  <TD>Date/Time</TD>
  <TD>File Size</TD>
  <TD>Dimensions</TD>
</TR>
{EXIF-FILE}
<TR>
  <TD><a href="{file-name-full}">{file-name}</a></TD>
  <TD>{date-taken}</TD>
  <TD>{file-size}</TD>
  <TD>{width}x{height}</TD>
</TR>
{EXIF-TAIL}
</TABLE>
</HTML>
```

Sample output from the above command and template run on a directory containing two files could be as follows:

```
<HTML>
<HEAD>My Photographs</HEAD>
<TABLE ID="Table1" BORDER=1 CELLSPACING=3 CELLPADDING=1 WIDTH=725>
<TR>
  <TD>File Name</TD>
  <TD>Date/Time</TD>
  <TD>File Size</TD>
  <TD>Dimensions</TD>
</TR>
<TR>
  <TD><a href="mydir/photo1.jpg">photo1.jpg</a></TD>
  <TD>2003:01:01 11:35:58</TD>
  <TD>65848</TD>
  <TD>1760x1168</TD>
</TR>
<TR>
  <TD><a href="mydir/photo2.jpg">photo2.jpg</a></TD>
  <TD>2003:01:01 11:40:00</TD>
  <TD>63458</TD>
  <TD>1760x1168</TD>
</TR>
</TABLE>
</HTML>
```

Example 4

You want to display one line per file, starting with the file name, a colon, then a list of fields separated by semicolon characters. A template to produce this out would look like this:

```
exiflist -o t -s “{file-name}: {make};{model};{shutter};{f-number}\n” .
```

Sample output from this command is:

```
photo1.jpg: NIKON;E995;1/125;2.8
photo2.jpg: NIKON;E995;1/64;5.6
photo3.jpg: NIKON;E995;1/125;2.8
```

8 Command Reference

8.1 EXIFCOPY

8.1.1 Description

In its simplest form the `exifcopy` command is used to copy Meta data unmodified from one image file (the *source* file) to another image file (the *destination* file). For example:

```
exifcopy source.jpg dest.jpg
```

By default a backup copy of the destination file is kept. The name of the backup file will contain the suffix '`-be`' appended to the file name immediately before the '.' in the file name. For example, if the destination file name was `myphoto.jpg`, the backup file will be called `myphoto-be.jpg`. If you do not want a backup file, use the `-b` option.

If the destination file already contains Meta data, `exifcopy` will refuse to copy any Meta data. To force `exifcopy` to overwrite any Meta data in the destination file, use the `-o` option. This will cause any Meta data in the source file to overwrite Meta data in the destination file. If a particular type of Meta data is not present in the source file, the corresponding data in the destination file will be left unchanged.

For example, if the `image1.jpg` contains only EXIF data and the `image2.jpg` contains only IPTC data, then the command:

```
exifcopy -o image1.jpg image2.jpg
```

will copy the EXIF data from `image1.jpg` into `image2.jpg`, and will leave the IPTC data in `image2.jpg` unchanged[12].

8.1.1.1 Advanced Use

`exifcopy` can be used to copy Meta data between multiple image files in one command. This is achieved by specifying:

- both the source and the destination as directory names
- the source as a single file and the destination as a directory name

The selection of the source and destination files between which Meta data will be copied is controlled by the `-m` (match) option. The `-m` option specifies how many characters, counting from the left of the source file name, must be the same in the destination file name before copying will occur; if the file names match for the specified number of characters then the EXIF data will be copied between the files.

The `-m` option can be specified in two ways:

1. as the number of characters that must match (eg. `-m 8`)
2. as a dot (`-m .`) indicating that the files much match up to the first '.' in the source file name. For example, suppose the source file is `myphoto.jpg` and the possible destination files are `myphoto1.jpg` and `mypic.jpg`. Meta data will only be copied to `myphoto1.jpg` because there are seven characters before the '.' in the source file name, and `myphoto1.jpg` matches the source file name for the first seven characters.

If no `-m` option is specified then the file names must match exactly for the Meta data to be copied.

[12] If you want to separately delete EXIF, IPTC, XMP, or Flashpix information from an image file, use the `exifedit` command with the `/e` option. See Section 8.3 for more details.

Specifying a match length of zero (`-m 0`) causes all sources files to match all destination files. This is only useful when the source is a single file and the destination is a directory. In this scenario, specifying `-m 0` causes the meta data from the single source file to be copied into all the JPG files in the destination directory.

8.1.1.2 Supported Image File Types

For some file types it is possible to make an exact copy of all EXIF, IPTC, XMP, and Flashpix data from one file to another. For other file types only fields that can be edited[13] by the EXIFutils can be copied between files.

Table 22 shows the image file types that support exact copying of all Meta data. Note that Minolta MRW files can only be the source file for an exact copy operation, not the destination file.

Table 22 File types for which exact copy is supported

File Type	**Exact copying of all EXIF fields supported**	**Exact copying of all IPTC fields supported**	**Exact copying of all Flashpix data supported**
Adobe .DNG			
JPEG	✓	✓	✓
TIF			
EXIF .EXI	✓	✓	✓
Canon .CRW			
Canon .CR2			
EPS			
Fujifilm .RAF			
Kodak .DCR			
Minolta .MRW	✓ (Note 1)		
Nikon .NEF			
Olympus .ORF			
Sony .ARW			

Note 1: An .MRW file can only be the *source* of the copied fields. Copying of field *into* an .MRW file is not supported

If both source and destination files support exact copying then by default an *exact* copy of *all* Meta data in the source file will be copied into the destination file. If the destination file already contains Meta data, `exifcopy` will not overwrite it unless the `-o` option (force overwrite) option is specified. If the `-f` option is specified then only the fields listed in the `-f` option will be copied (see below).

Performing an exact copy of all fields is not supported if the source file, destination file, or both do not support exact copying. In these cases only fields that EXIFutils considers to be editable[14] can be copied[15].

If you do not want to make an exact copy of all fields, or if exact copying is not supported for your file type, you can copy individual fields using the `-f` option. The `-f` option is used to specify a list of the nicknames of the fields to be copied, eg:

```
-f make,model,description,shutter
```

[13] See Section 8.3.1.1 “Editable and non-Editable Fields” for an explanation of editable fields.
[14] See EXIFutils Field Reference Guide for a list of fields that are editable.
[15] You can force `exifcopy` to copy only selected fields between files that support exact copying by specifying the `/f` option.

If the `-f` is not specified then by default all `exif-common` and `iptc` fields are copied. See Section 4.5 “Selecting Multiple Fields” for a description of the `exif-common` and `iptc` nicknames.

`exifcopy` does not check that the Meta data being copied is consistent with the image file into which it is being copied.

8.1.1.2.1 Canon Raw (.CRW) Files

Canon Raw .CRW files do not contain EXIF data; they contain Canon proprietary Meta data. If a .CRW files is the source of an `exifcopy` operation then the .CRW Meta data will be copied into the equivalent EXIF field in the destination file. If there is no equivalent EXIF field then the .CRW field is not copied. Table 23 shows the equivalent EXIF field for each .CRW field. Note that in some cases there is more than one EXIF field corresponding to a Canon .CRW field. In these cases the Canon field will be copied into all equivalent EXIF fields. For example, the Canon `crw-shutter` field value will be copied into both the `exp-time` and `shutter` EXIF fields.

A Canon .CRW file cannot be the destination of an `exifcopy` operation.

Table 23 Canon .CRW to EXIF Field Translations

Canon .CRW Field	Equivalent EXIF Field	Description
`crw-vendor`	`make`	Camera Make
`crw-model`	`model`	Camera Model
`crw-date`	`date-taken`	Date Taken
`crw-firm-ver`	`firm-ver`	Firmware Version
`crw-iso`	`iso-speed`	ISO Speed Setting
`crw-exp-comp`	`exp-bias`	Exposure Compensation
`crw-shutter`	`shutter`	Shutter Speed
	`exp-time`	Exposure Time
`crw-aperture`	`aperture`	Aperture
	`f-number`	F-number

8.1.2 Command Syntax

```
exifcopy [-b]           [-o] [-q] [-f field-list] \
                                  [-p cmd=charset,exif=charset,iptc=charset] \
                                                           src-file dest-file

exifcopy [-b] [-c] [-I] [-o] [-q] [-f field-list] \
                                  [-p cmd=charset,exif=charset,iptc=charset] \
                                                     src-file dest-directory

exifcopy [-b] [-c] [-I] [-o] [-q] [-f field-list] \
                                  [-p cmd=charset,exif=charset,iptc=charset] \
                                                src-directory dest-directory
exifcopy [-v|-h]
```

`-b` Do not save a backup copy of the original destination file(s).

`-c` File name comparisons are to be case-sensitive (this is the default on Linux, MacOS X, and Solaris).

`-f` *`field-list`*
Copy only the fields specified in *`field-list`* from the source to the destination file(s).

`-h` Display `exifcopy` usage information

`-i` File name comparisons are to be case-insensitive (this is the default on Windows)

`-m [.|n]` Specifies the number of characters that must match between the source file name and destination file name before Meta data will be copied. '`-m .`' specifies that the file names must match up to the position of the first '.' in the source file name.

`-o` Force overwriting of EXIF data in the destination file.

`-p [cmd=xxx],[exif=xxx],[iptc=xxx]`
Specify the character sets to be used when interpreting text fields
`cmd=xxx` sets the charset to be used for output displayed from this command.
`exif=xxx` sets the charset to be used for reading/writing EXIF text fields
`iptc=xxx` sets the charset to be used for reading/writing IPTC text fields

Any of the three may be omitted. For example:

```
-p exif=latin1,iptc=mac
-p cmd=utf8
```

These values can also be set using the `EXIFUTILS_CS_CMD`, `EXIFUTILS_CS_EXIF`, `EXIFUTILS_CS_IPTC` environment variables.

See Section 6 "Character Sets" for more information.

`-q` Quiet mode. Do not display warning messages.

`-v` Display program version information. This is the default action if no other options are present.

`/?` Display `exifcopy` usage information. This is the same as the `/h` option.

8.1.3 Examples

```
exifcopy Dcp_0123.jpg Dcp_0123_new.jpg
```

Takes an exact copy of all Meta data (EXIF, IPTC, Flashpix) from `Dcp_0123.jpg` and copies it into `Dcp_0123_new.jpg`. An unaltered copy of `Dcp_0123_new.jpg` will be left in the current directory with the name `Dcp_0123_new-be.jpg`. If `Dcp_0123_new.jpg` already contains Meta data then an error will be displayed and `Dcp_0123_new.jpg` will be left unaltered.

```
exifcopy -o -b Dcp_0123.jpg Dcp_0123_new.jpg
```

Copies the Meta information from `Dcp_0123.jpg` into `Dcp_0123_new.jpg`. If `Dcp_0123_new.jpg` already contains Meta information then it will be overwritten (`-o` option). No backup copy of `Dcp_0123_new.jpg` will be kept (`-b` option).

```
exifcopy -f date-taken,f-number,ip-caption Dcp_0123.jpg Dcp_0123.tiff
```

Copies the `date-taken` and `f-number` EXIF fields, and the `ip-caption` IPTC field from `Dcp_0123.jpg` into `Dcp_0123.tiff`. If `Dcp_0123.tiff` already contains any of these fields then they will be overwritten. An unaltered copy of `Dcp_0123.tiff` will be left in the current directory with the name `Dcp_0123-be.tiff`.

```
exifcopy PICT0123.mrw PICT0123.jpg
```

Takes an exact copy of all Meta information from `Dcp_0123.mrw` and copies it into `Dcp_0123.jpg`. An unaltered copy of `Dcp_0123.jpg` will be left in the current directory with the name `Dcp_0123-be.jpg`. If `Dcp_0123.jpg` already contains Meta information then an error will be displayed and `Dcp_0123.jpg` will be left unaltered.

8.2 EXIFDATE

8.2.1 Description

The `exifdate` command modifies all EXIF date/time fields in the specified image files. If the specified file is a directory then the date fields in all image files in that directory will be modified.

The way in which the date is modified is governed by the action parameter. Valid actions are:

- Add (`-a`) the specified date/time value will be *added* to every EXIF date/time field
- Subtract (`-s`) the specified date/time value will be *subtracted* from every EXIF date/time field
- Replace *replace* some or all of the EXIF date/time with a specific value. This is the default action if neither `-a` or `-s` is specified.

The value to *add to*, *subtract from*, or *replace with* is specified by the date/time parameter.

The format of the date/time parameter is:

```
yyyy:mm:dd hh:mm:ss
```

Note that when a full date/time is specified on the command line, quote characters must surround it. For example,

```
exifdate “2001:03:16 12:23:00” image.jpg
```

The quotes are required because the date/time contains an embedded blank.

Either the date or time portion can be omitted, so both of the following formats are also valid:

```
yyyy:mm:dd        eg. 2001:03:16
hh:mm:ss          eg. 01:00:00
```

If the action is *Replace*, any portion of the date or time can be replaced with x's to indicate that part of the field is not to be replaced. For example, in the command

```
exifdate 1999:xx:xx image.jpg
```

the date parameter indicates that the year field is to be set to 1999, but the month and day are to be left unchanged. The time will also be left unchanged because the time part of the date/time was not specified.

8.2.1.1 Supported Image File Types

Table 24 shows the files types that are supported by `exifdate`.

Table 24 File Types Support by exifdate

File Type	Supported by exifdate
Adobe .DNG	✓
JPEG	✓
TIFF	✓
EXIF .EXI	✓
Canon .CRW	
Canon .CR2	✓
EPS	
Fujifilm .RAF	✓
Kodak .DCR	✓
Minolta .MRW	✓
Nikon .NEF	✓
Olympus .ORF	✓
Sony .ARW	✓

8.2.2 Command Syntax

```
exifdate -a [-q] [-r] date-time file,... directory,...
exifdate -s [-q] [-r] date-time file,... directory,...
exifdate    [-q] [-r] date-time file,... directory,...
exifdate [-h|-v]
```

`-a` Add *`date-time`* to all EXIF date fields in the specified image files.

`-h` Display `exifdate` usage information.

`-q` Quiet mode. Do not display warning messages.

`-r` Include subdirectories. If any directories are specified, then dates in all image files in the specified directory and all of its subdirectories will be modified.

`-s` Subtract *`date-time`* from all EXIF date fields in the specified image files.

`-v` Display program version information. This is the default action if no other options are present.

`-?` Display `exifdate` usage information.

8.2.3 Examples

```
exifdate "1999:06:01 14:27:08" image1.jpg image2.jpg
```

Set the date fields in `image1.jpg` and `image2.jpg` to `1999:06:01 14:27:08`

```
exifdate "xxxx:06:xx" my-dir1 mydir2
```

Change the month to 06 in all date fields in image files in the directories `my-dir1` and `mydir2`.

```
exifdate -a "0001:03:05" image.jpg
```

Add 1 year, 3 months, and 5 days to all date/time fields in `image.jpg`.

```
exifdate -a "01:30:00" image.jpg
```

Add 1 hour and 30 minutes to all date/time fields in `image.jpg`.

```
exifdate -r -s "12:00:00" my-dir
```

Subtract 12 hours from the date/time fields of all image files in directory `my-dir` and all of its subdirectories (`-r` option).

8.3 EXIFEDIT

8.3.1 Description

The `exifedit` command adds, modifies, and deletes meta data fields from image files. The edit action to be performed is specified by one or more of the following command line options:

`-a` Add one or more fields. The fields to be added, and their values, are specified in a field value list following the `-a` option. Fields already existing in the image file will have their value replaced with the specified value. See Section 8.3.1.2 for more detail.

`-r` Remove one or more fields. The fields to be removed are specified as a field list following the `-r` option. See Section 8.3.1.3.

`-s` Update the EXIF `width` and `height` fields to match actual image values (JPEG files only). See Section 8.3.1.4.

`-t` Remove or update EXIF thumbnail image. See Section 8.3.1.4.

`-e` Erase all EXIF, IPTC, XMP and/or Flashpix information from the file (JPEG and .EXI files only). See Section 8.3.1.5.

By default a copy of the original (unedited) image file will be saved with a '`-be`' suffix. For example, if you edit the file `myphoto.jpg` then a backup will be saved with the name `myphoto-be.jpg`. Use the `-b` option if you do not want the backup file created.

8.3.1.1 Editable and non-Editable Fields

Editable fields are fields for which EXIFutils can change the value. EXIFutils does not allow the user to edit all meta data fields for one or more of the following reasons:

1. The value of some fields must be generated by EXIFutils to ensure correct formatting of the meta data structures.
2. EXIFutils does not yet support formatting of the data type in which the field is stored.
3. The field is a non-standard field that EXIFutils does not know how to correctly format it. This is most often the case with Maker Note fields, which are defined by each camera manufacturers and for which the correct format definitions are not published.

A complete list of all fields, with an indication of whether they are editable by EXIFutils is available in the EXIFutils Field Reference Guide.

8.3.1.2 Adding Individual Meta Data Fields (`-a` option)

The `exifedit -a` option is used to add EXIF and IPTC fields to an image file. To use `exifedit` you need to know the nickname of the field you want to add, and the value to which you want to set it. See EXIFutils Getting Started Guide for an explanation of how to find the nickname of a field.

Once you know the nickname and the value you want to set, enter the following command:

```
exifedit -a nickname=value file-name
```

For example, if you want to add the shutter speed field (nickname `shutter`) with a value of 1/125 to `myphoto.jpg`, enter the following command:

```
exifedit -a shutter=1/125 myphoto.jpg
```

Note that if the field value contains a space character, or if you include one or more spaces before or after the “=”, then the whole nickname=value string must be surrounded by quotes, eg:

```
exifedit -a “date-taken=2004:01:01 23:59:59” myphoto.jpg
```

Also, if the field value contains a comma or a semicolon, then it is necessary to put quotes[16] around both the field value and the whole *nickname=value* string. To do this you must escape the embedded quote characters[17]. For example:

```
exifedit -a “comment=\”I took this, for sure\”” myphoto.jpg
```

To set multiple EXIF and/or IPTC fields in one command, you can specify mutltiple nickname/value pairs in the -a option, separated by commas, i.e.

```
exifedit -a nickname=value,nickname=value,nickname=value file-name
```

As before if there are spaces anywhere in the nickname=value string, or if a field value contains a comma or a space, then the whole string must be surrounded by quotes, eg:

```
exifedit -a “shutter=1/125, comment=\”I took this, for sure\”” myphoto.jpg
```

For a description of how to add more complex field types such as XMP structure fields and list fields, see Section 4.2 “Nicknames for XMP Structure and List Fields’.

8.3.1.2.1 Adding Repeating IPTC Fields

Most IPTC fields can only appear once in an image file. For example, you can only put one `ip-caption` field in an image. If you add a caption field when one is already present, EXIFutils will delete the existing `ip-caption` field and replace it with the new caption.

Some IPTC fields can occur more than once, for example the Keyword field (nickname `ip-keyword`) and Supplementary Category field (nickname `ip-suppcat`). Repeating IPTC fields can appear up to 999[18] times within an image file. See the EXIFutils Field Reference Guide for information about which IPTC fields can be present more than once.

Each time a repeating field is added to a file, a new instance of that field is added to the file. For example, if you run the following two commands one after the other, there will be two `ip-keyword` fields in the file:

```
exifedit   -a ip-keyword=sports myfile.jpg
exifedit   -a ip-keyword=soccer myfile.jpg
```

When you display a repeating field with `exiflist`, it will display all of the instances that are in the file. For example:

```
>exiflist -o ln -f ip-keyword myfile.jpg
ip-keyword="sport; soccer"
```

You can add multiple repeating values in one command by using one of the following formats:

```
exifedit -a ip-keyword=sports,ip-keyword=soccer myfile.jpg
exifedit -a “ip-keyword=sports;soccer” myfile.jpg
```

[16] See also the `exifedit /c` option, which can be used to change the quote character used when quoting `exifedit` field values.

[17] The examples given in this section apply to the Windows version of EXIFutils and to the OSX and Linux versions of the product when used with a Bourne-based command shell. The method used to embedded quotes when using other command shells may be different.

[18] The limit of 999 repetitions is imposed by EXIFutils. According to the IPTC specification the number of times a field can be repeated is limited only by available memory.

These two commands have exactly the same result, and each is equivalent to the two commands shown in the initial `exifedit` example above.

By default, a semicolon is used to separate multiple repeating field values. You can change this separator character using the `-j` option.

8.3.1.2.2 Setting Field Values Using the Value of Other Fields

Using *template* values you can set fields using the values of other fields. For example, if you have a description of a photo in the EXIF `description` field, and you want to copy this value into the IPTC `ip-caption` field, you can do the following:

```
exifedit -a “ip-caption={description}” myfile.jpg
```

You can also combine multiple fields into one value, for example:

```
exifedit -a “ip-caption=Photo taken at {date-taken}, F-stop {f-number}” myfile.jpg
```

See also Section 5.4 “Converting Between EXIF, XMP, and IPTC Date/Time Formats” for more information on setting EXIF date/time fields based on IPTC date/time values, and vice versa.

8.3.1.3 Removing Individual Fields (`-r` option)

The `exifedit -r` option is used to remove individual meta data fields from an image file. To do this you need to know the nickname of the field you want to remove. See EXIFutils Getting Started Guide for an explanation of how to find the nickname of a field.

Once you know the nickname of the field(s) you want to remove, enter the following command:

```
exifedit -r nickname,nickname,nickname file-name
```

For example, to remove the EXIF description field, and the IPTC caption field, enter the following command:

```
exifedit -r description,ip-caption myfile.jpg
```

To remove all EXIF and/or IPTC fields, see Section 8.3.1.5 “Erasing all Meta data (-e option)”.

For a description of how to remove more complex field types such as XMP structure fields and list fields, see Section 4.2 “Nicknames for XMP Structure and List Fields’.

8.3.1.3.1 Removing Repeating IPTC Fields

As explained in Section 0, some IPTC fields can be present multiple times in an image file. If you remove any of these fields using the `exifedit -r` option, all occurrence of the field will be removed.

For example, if the file `myfile.jpg` contains two occurrences of the `ip-keyword` field added by the command:

```
exifedit -a “ip-keyword=planes,ip-keyword=planes” myfile.jpg
```

then the following command will delete both occurrences of the `ip-keyword` field:

```
exifedit -r ip-keyword myfile.jpg
```

EXIFutils does not provide the option to delete individual occurences of a repeating IPTC field.

8.3.1.4 Updating EXIF fields to match the actual image (`-s` and `-t` options)

Some photo editing programs do not update the EXIF image information when you save an image after editing it. EXIFutils provides the `exifedit -s` and `-t` options update the EXIF fields.

The `-s` option is used to set the EXIF image `width` and `height` values to the actual size of the main image. The `-t` option is used update the EXIF thumbnail image to match the main image. The `-s` and `-t` options can be used either together or separately.

For example:

```
exifedit -s myfile.jpg
```

```
exifedit -t a myfile.jpg
```

```
exifedit -s -t a myfile.jpg
```

The `exifedit -t` a option creates a thumbnail with the default dimensions of 160x120. See Section 8.3.2 for details of how to change the size of the generated thumbnail.

8.3.1.5 Erasing all Meta data (`-e` option)

The `exifedit -e` option is used to delete all Meta data from a JPEG image file. It can also be used to selectively remove only EXIF, IPTC, XMP, or Flashpix meta data from a JPEG file.

To delete all Meta data (EXIF, IPTC, and Flashpix) use the '`-e a`' option. For example:

```
 exifedit -e a myfile.jpg
```

To delete all EXIF Meta data, but leave IPTC, XMP and Flashpix Meta data, use the '`-e e`' option. For example:

```
exifedit -e e myfile.jpg
```

To delete all IPTC Meta data, but leave EXIF, XMP and Flashpix Meta data, use the '`-e i`' option. For example:

```
exifedit -e x myfile.jpg
```

To delete all XMP Meta data, but leave EXIF, IPTC and Flashpix Meta data, use the '`-e x`' option. For example:

```
exifedit -e i myfile.jpg
```

To delete all Flashpix Meta data, but leave EXIF, IPTC and XMP Meta data, use the '`-e f`' option. For example:

```
exifedit -e f myfile.jpg
```

Deletion of all Ricoh RMETA Meta data is not supported.

8.3.1.6 Supported Image File Types

Table 25 shows the image file types supported by `exifedit`.

Table 25 Image File Types Supported by exifedit

File Type	-e option	-s option	All other options
JPEG	✓	✓	✓
EXIF .EXI	✓		✓
TIF			✓
Adobe .DNG			✓
Canon .CR2			✓
Kodak .DCR			✓
Nikon .NEF			✓
Canon .CRW			
Fujifilm .RAF			
Minolta .MRW			
EPS			
Olympus .ORF			
Sony .ARW			

8.3.2 Command Syntax

```
exifedit  [-b] [-q] [-i] [-s] [-u] [-c x] [-j x] \
                             [-p cmd=charset,exif=charset,iptc=charset] \
                             [-a field-value-list|@filename] \
                             [-r field-list] \
                             [-t [r|a|a,size|a,wwwxhhh|a,@filename]] file...
exifedit [-b] [-q] [-i] -e [a|e|f|i] file...
exifedit [-h|-v]
```

`-a [field-value-list | @file-name | @-]`

Add the specified field(s) to the image file(s). The *field-value-list* is specified as a comma separated list of field nickname/value pairs in the following format:

```
-a  fieldname=fieldvalue,fieldname=fieldvalue
```

For example:

```
-a “description=I took this,shutter=1/125,f-number=2.8”
```

The *fieldvalue* can also be specified as a template. See Section 7 “Templates” for more details.

Alternatively, the name of a file containing the *field-value-list* in the above format can be specified. For example:

```
-a @my-field-values.txt
```

The file name can also be specified as ‘-‘ which indicates that the *field-value-list* will be read from stdin. For example:

```
-a @-
```

`-b` Do not keep a backup copy of the original unedited image file(s).

`-c` *x* Specifies the character to be used as the quote character in the field value list. ‘*x*‘ can be any printable character except “ or \. For example

```
-c : -a “description=:My birthday, last year:”
```

sets the quote character to the colon character and uses it to delimit the text value of the description field.

`-e [a|e|f|i]`

Erase Meta data from the file. The types of Meta data to be removed are specified by the sub-parameter:

`-e a` Erase all EXIF, IPTC, XMP, and Flashpix Meta data.
`-e e` Erase all EXIF data. All APP1 EXIF data blocks will be removed.
`-e i` Erase all IPTC data. All APP13 data blocks will be removed.
`-e x` Erase all XMP data. All APP1 XMP data blocks will be removed.
`-e f` Erase all Flashpix data. All APP2 FlashPix data blocks will be removed.

This option is only valid for JPEG and .EXI files. It has no effect on other file types. This option cannot be specified with the `-a, -r, -t,` or `-s` options.

`-h` Display `exifedit` usage information.

`-i` Include subdirectories. If any of the files specified is a directory, then image files in the specified directory and all of its subdirectories will be processed.

CAUTION SHOULD BE TAKEN WHEN USING THIS OPTION AS IT CAN POTENTIALLY MODIFY A LARGE NUMBER OF FILES.

`-j [`*x*`|`*0xnn*`|`*0nnn)*

Specify the field separator character to be used to separate repeating IPTC fields in the `-a` option. This can be specified in one of three ways:

- As a printable character, for example `-j +`
- As a hexadecimal value, for example `-j 0x10`
- As an octal value with a leading zero, for example `-j 020`

The default value is a semicolon ‘`;`’

`-p [cmd=xxx],[exif=xxx],[iptc=xxx]`

Specify the character sets to be used when interpreting/displaying text fields

`cmd=xxx` sets the charset to be used for output displayed from this command.
`exif=xxx` sets the charset to be used for reading/writing EXIF text fields
`iptc=xxx` sets the charset to be used for reading/writing IPTC text fields

Any of the three may be omitted. For example:

```
-p exif=latin1,iptc=mac
-p cmd=utf8
```

These values can also be set using the `EXIFUTILS_CS_CMD`, `EXIFUTILS_CS_EXIF`, `EXIFUTILS_CS_IPTC` environment variables.

See Section 6 “Character Sets” for more information.

`-q` Quiet mode. Do not display warning messages

`-r` *`field-list`*

Remove the specified meta data fields. The field list is specified as a comma separated list of field nicknames in the following format:

```
-r  fieldname,fieldname,fieldname,fieldname
```

For example:

```
-r description,shutter,f-number,comment,xex-flash.fl-fired
```

Alternatively, the name of a file containing the *`field-list`* in the above format can be specified. For example:

```
-r @my-field-list.txt
```

The file name can also be specified as dash ‘`-`‘ which indicates that the *`field-list`* should be read from stdin. For example:

```
-r @-
```

`-s` Set the EXIF `width` and `height` fields for the main image to match the actual width and height of the main JPEG image.

This option can only be used on JPEG files. It has no effect on other file types.

`-t [a|a,`*`size`*`|a,`*`ww`*`x`*`hh`*`|a,@`*`filename`*`|r]`

The ‘`-t a`’ option replaces an existing thumbnail image (if any) with either a thumbnail provided by the user, or one generated from the main JPEG image by `exifedit`. If a thumbnail is to be generated, the size of the generated thumbnail image can be specified in one of three ways:

`-t a` The generated thumbnail will be the default size 160 pixels wide by 120 pixels high. This size conforms to the DCF V1.0 specification and therefore many digital cameras and image processing programs expect the thumbnail to be this size.

`-t a,`*`size`* The specified *`size`* is the size (in pixels) of the longest dimension of the thumbnail image. The size of the other dimension will be scaled to maintain the aspect ratio of the image. For example, if you specified `-t a,100` and the size of the main image is 1000x500, then the thumbnail dimensions will be 100x50. Using the same thumbnail specification `-t a,100` with a main image size of 500x1000, then the thumbnail dimensions will be 50x100.

`-t a,`*`ww`*`x`*`hh`* The thumbnail dimensions will be *ww* pixels wide by *hh* pixels high, eg `-t a,200x100`

If the thumbnail is to be replaced with a JPEG image provided by the user, then it is specified as follows:
`-t a,@`*`filename`*

The thumbnail will be replaced with the image in the specified `filename`. The specified thumbnail file must be in the same directory as the image file, and is checked to ensure that it is a JPEG file.

The `-t r` option removes any existing EXIF thumbnail image and all thumbnail related EXIF fields.

The `-t` option can only be used on JPEG files. It has no effect on other file types.

`-u` Store EXIF `comment` field as UNICODE (default is ASCII). See Section 6.3 “EXIF Comment Field” for more information.

`-v` Display program version information. This is the default action if no other options are present.

`-y` Used when invoking this command from a .BAT file that has been created with an ANSI text editor. This option translates the command parameters so that non-English characters are interpretted correctly. If present this option MUST be the first option on the command.

`-?` Display `exifedit` usage information.

8.3.3 Examples

```
exifedit -a "copyright=Copyright (c) 2001 John Citizen" my-dir
```

Add the `copyright` field to all image files in the directory `my-dir`. If the copyright field is already present in any of the files, it will be replaced with the value specified. A backup copy of all edited files will be created.

```
exifedit -r comment,xex-flash.fl-fired,ip-keyword myfile.jpg
```

Removes the `comment`, `xex-flash.fl-fired`, and all instances of the `ip-keyword` field from `myfile.jpg`.

```
exifedit -t r mydir fred.jpg
```

Removes the thumbnail images and EXIF Thumbnail fields from all the JPEG files in the directory `mydir`, and from the file `fred.jpg` in the current directory.

```
exifedit -t a,@fred-thumb.jpg mydir fred.jpg
```

Replaces the thumbnail image in `fred.jpg` with the image in `fred-thumb.jpg`.

```
exifedit -c : -a “comment=:my photo:” -t a -r copyright mydir
```

Performs the following changes to all JPEGs in the `mydir` directory:

- adds the comment field with the value `'my photo'` (`-a` option)
- replaces any existing EXIF thumbnail image with a new thumbnail based on the main JPEG image. (`-t` option)
- removes the copyright field (`-r` option)

An alterative quote character ':' is used (`-c` option).

```
exifedit -a @fields.txt myfile.jpg
```

Adds the fields/values specified in the file `fields.txt` (located in the current directory) into the JPEG file `myfile.jpg`

Sample contents of `fields.txt` is show below:

```
copyright="Copyright (c) Jane Citizen.",
description="Loch Ness in winter",
comment="This is a photo I look of Loch Ness in winter.
The comment spans more than one line, but that is OK
because exifedit will replace each new line character
with a space."
```

```
exifedit -a “comment=Taken at {date-taken} by {ip-byline}” myfile.jpg
```

Adds the `comment` field to `myfile.jpg`. The value of the comment field is generated using a supplied substitution template. If the `date-taken` field has a value of ‘`2000:12:31 23:59:59`’ and the `ip-byline` field has value of “`Jane Smith`”, then in the above example the comment field will be set to:

```
“Taken at 2000:12:31 23:59:59 by Jane Smith”.
```

```
exifedit -a “ip-headline=Joe Triumphs,ip-keyword=sport,keyword=olympics” myfile.jpg
```

Adds the IPTC `ip-headline` field, and two instances of the IPTC `ip-keyword` field to `myfile.jpg`.

8.4 *EXIFFILE*

8.4.1 Description

The `exiffile` command is used to change the attributes of an image file based on the value of meta data field values. The operation to be performed by `exiffile` is selected by one of the following command line options:

- `-n` Rename one or more image files based on field values
- `-t` Change the "last modified" timestamp of one or more images files to match the date/time that each image was taken.

Both the `-n` and `-t` options can be specified on the same command.

8.4.1.1 Renaming Files

`exiffile` renames image files based on a file name *template* specified in the `-n` option. A template is a text string containing markers indicating where field values are to be substituted. A marker consists of an EXIFutils field nickname surrounded by curly brackets "`{ }`". For example, if you wanted to rename an image file so that the file name was the same as the date/time the image was taken, then you would use the following command:

```
exiffile -n "{date-taken}.jpg" myphoto.jpg
```

You can use any EXIFutils field nickname in the template. See the EXIFutils Field Reference Guide for a complete list of nicknames. Templates are described in more detail in Section 7 "Templates".

To ensure that the resulting file name is valid, `exiffile` will replace any illegal characters with a dash "-". You can change this substitution character to a different character using the `-c` option. The only character that is illegal in a file name is '/'

If you specify the `-s` option, `exiffile` will also replace any space characters in the file name with the substitution character.

8.4.1.1.1 File Name Clashes

When attempting to rename a file, `exiffile` first checks to see if a file with that name already exists. If it does, then `exiffile` will add a suffix to the name to ensure that the name is unique. The suffix is a number surrounded by brackets, eg "`(1)`". For example, if a file is to be renamed to "`2004-01-01-12-23-23.jpg`", but a file with that name already exists, `exiffile` will rename the file to "`2004-01-01-12-23-23(1).jpg`". If that file name also exists, `exifile` will attempt to rename it to "`2004-01-01-12-23-23(2).jpg`", and so on.

8.4.1.2 Changing the "Last Modified" Date

When the `-t` option is specified `exiffile` will, for each image, attempt to set the last modified date of the file to be the same as the value of the `date-taken` field. If the `date-taken` field is not present in the image file then the last modified timestamp will be set to the value of the `date-digi` (date digitised) field. If the `date-digi` field is not present in the image file then the last modified timestamp will be set to the value of the `date-mod` (date modified) field.

If none of the above date fields is present in the image file then an error is displayed and no change is made to the last modified date of the file.

Note: `exiffile` attempts to adjust for Daylight Saving Time (DST). To do so `exiffile` relies on the underlying settings and services provided by the operating system. Correct Daylight Saving/Summer time conversion relies on these system settings being correct.

8.4.1.3 Supported Image File Types

Table 26 shows the image file types supported by `exiffile`.

Table 26 Image File Type Supported by exiffile

File Type	-n option	-t option
JPEG	✓	✓
EXIF .EXI	✓	✓
TIF	✓	✓
Adobe .DNG	✓	✓
Canon .CR2	✓	✓
Canon .CRW	✓	
EPS	✓	✓
Fujifilm .RAF	✓	✓
Minolta .MRW	✓	✓
Kodak .DCR	✓	✓
Nikon .NEF	✓	✓
EPS	✓	✓
Olympus .ORF	✓	✓
Sony .ARW	✓	✓

8.4.2 Command Syntax

```
exiffile  [-q] [-r] [-t] [-c x] [-s][-p cmd=charset,exif=charset,iptc=charset]\
                  [-n file-name-template|@filename|@-] file...
exiffile [-h|-v]
```

`-c x` Sets the character to be used to replace illegal characters (and optionally space characters – see `-s` option) in file names. If `-c` is not specified then by default the dash character “-“ will be used. This option is ignored if the `-n` option is not also specified.

`-h` Display `exiffile` usage information.

`-n [`*`file-name-template`*` | `*`@filename`*` | `*`@-`*`]`

Rename the image file to the name specified by *`file-name-template`*. See Section 7 “Templates” for more information about templates. This option can be specified in one of three ways:

1) on the command line, for example:

```
-n {date-taken}-{model}.jpg
```

2) as an ampersand followed by the name of a text file containing the *`file-name-template`*:

```
-n @template.txt
```

3) an ampersand followed by a '-', indicating that the *`field-name-template`* is to be read from stdin, for example:

```
-n @-
```

After substitution of field values in to the *`file-name-template`* is complete, any characters that are invalid in a file name will be replaced with the substitution character (see `-c` option). If `-s` has been specified, any spaces in the file name will also be replaced with the substitution character.

-p [cmd=xxx],[exif=xxx],[iptc=xxx]

Specify the character sets to be used when interpreting/displaying text fields

cmd=xxx	sets the charset to be used for output displayed by this command.
exif=xxx	sets the charset to be used for reading EXIF text fields
iptc=xxx	sets the charset to be used for reading IPTC text fields

Any of the three may be omitted. For example:

```
-p exif=latin1,iptc=mac
-p cmd=utf8
```

These values can also be set using the EXIFUTILS_CS_CMD, EXIFUTILS_CS_EXIF, EXIFUTILS_CS_IPTC environment variables.

See Section 6 “Character Sets” for more information.

-q Quiet mode. Do not display warning messages.

-r Include subdirectories. If any of the files specified is a directory, then image files in the specified directory and all of its subdirectories will be processed.

-s Replace spaces. Any spaces in the file name specified in the -n option will be replaced with the substitution character (see -c option). This option is ignored if the -n option is not also specified.

-t Set last modified timestamp of the image file to match the value in the date-taken EXIF field.

-v Display program version and licensing information. This is the default action if no other options are present.

-y Used when invoking this command from a .BAT file that has been created with an ANSI text editor. This option translates the command parameters so that non-English characters are interpretted correctly. If present this option MUST be the first option on the command.

-? Display exiffile usage information.

8.4.3 Examples

```
exiffile -t myimage.jpg
```

Sets the last modified timestamp of the image file myimage.jpg to the value of the date-taken field.

```
exiffile -t -r .
```

Sets the last modified timestamp of all image files in the current directory and any subdirectories of the current directory to the value of the date-taken field.

```
exiffile -n {date-taken}-{description}.jpg myphoto.jpg
```

Renames `myphoto.jpg` to the name resulting from the substitution of the `date-taken` and `description` EXIF fields. For example, if `date-taken` has a value of “2002:12:31 23:59:59”, and the `description` field has a value of “New Years Eve Party”, then `myphoto.jpg` will be renamed to:

```
2002:12:31 23:59:59-New Years Eve Party.jpg
```

```
exiffile -c _ -s -n {date-taken}-{description}.jpg myphoto.jpg
```

Rename `myphoto.jpg` to the name resulting from the substitution of the `date-taken` and `description` EXIF fields in the ***`file-name-template`***. Any characters that are illegal in a file name will be replaced with an underscore (`-c _` option). Any spaces will also be replaced with an underscore (`-s` option). For example, if `date-taken` has a value of “2002:12:31 23:59:59”, and the `description` field has a value of “New Years Eve Party”, then `myphoto.jpg` will be renamed to:

```
2002_12_31_23_59_59-New_Years_Eve_Party.jpg
```

8.5 EXIFINFO

8.5.1 Description

The `exifinfo` command displays information about EXIFutils settings and the about metadata fields that you can access, including:

- The names and valid values for metadata fields known to EXIFutils. See Section 8.5.1.1.
- EXIFutils option settings and environment variable values. See Section 8.5.1.2.
- The names of XMP schemas known to EXIFutils. See Section 8.5.1.3.

8.5.1.1 Field Attribute List Options

`exifinfo` is used to list details of the metadata fields that can be displayed, edited and deleted by the EXIFutils. Table 27 explains the format and values in the field list display.

The `-f` options is used to select the fields for which information will be displayed. You can display information about one or more specific fields, or about all fields of a certain type (eg. All IPTC fields). For example, the following command will display information about the XMP `xtif-res-unit` and `xex-scene-mode` fields:

```
exifinfo -f xtif-res-unit,xex-scene-mode

  ------- Field Name ------ -Attr- -Tag- ----- Description ----------

XMP TIFF 1.0
  xtif-res-unit              XRWS  0218  ResolutionUnit
             -----Value----- Abbrev --Num- ------Meaning-------
     Values: none            n          1  No Unit
             inch            i          2  Inch
             cm              c          3  Centimetre


XMP EXIF 1.0
  xex-scene-mode             XRWS  a406  SceneCaptureType
             -----Value----- Abbrev --Num- ------Meaning-------
     Values: standard        s          0  Standard
             landscape       l          1  Landscape
             portrait        p          2  Portrait
             night           n          3  Night Scene
```

If you only want to list the name and description of the fields (not their possible values), include the `-b` (brief) option:

```
exifinfo -b -f xtif-res-unit,xex-scene-mode

  ------- Field Name ------ -Attr- -Tag- ----- Description ----------

XMP TIFF 1.0
  xtif-res-unit              XRWS  0218  ResolutionUnit

XMP EXIF 1.0
  xex-scene-mode             XRWS  a406  SceneCaptureType
```

To display all IPTC use the following command:

```
exifinfo -f iptc
```

See Section 8.5.2 “Comand Syntax” for a list of valid values for the `-f` option.

Table 27 Explanation of Field List Information

<table>
<tr><th>Heading</th><th>Explanation</th></tr>
<tr><td>Field Name</td><td>The EXIFutils nickname for this field, which can be used to reference this field in EXIFutils commands.</td></tr>
<tr><td>Attr</td><td>Field Attributes. This contains four characters that provide information about the field.
<table>
<tr><td>1st character</td><td>Indicates whether this field is included in the exif-common or iptc group nicknames (Refer to EXIFutils Getting Started Guide). Possible values are:
‘E’ – this field is included in the exif-common group nickname
‘I’ – this field is included in the iptc group nickname
‘-‘ – this field is not included in either exif-common or iptc</td></tr>
<tr><td>2nd character</td><td>Indicates whether EXIFutils supports reading of this field value. Possible values are:
‘R’ – this field is readable
‘-’ – this field is not readable</td></tr>
<tr><td>3rd character</td><td>Indicates whether EXIFutils supports writing of this field. “Writing” means that this field can be individually edited by exifedit. Possible values are:
‘W’ – editing of this field is supported
‘-’ – editing of this field is not supported</td></tr>
<tr><td>4th character</td><td>Indicates whether this field is a “standard” field. Possible values are:
‘S’ – this field is defined in the relevant standard (see note 1).
‘-’ – this is not a standard field. Either it is a proprietary field defined by a camera maker, or is a field defined by EXIFutils.</td></tr>
</table></td></tr>
<tr><td>Tag</td><td>For EXIF and TIFF fields, this is the numeric tag that identifies the field. It is supplied for reference purposes only.

For IPTC fields, this value shows the dataset number and field number that together uniquely identify the IPTC field. They are show in the format x.y where ‘x’ is the dataset number and y is the field number.

The dataset number is used to group related IPTC fields together. Many Meta data editing applications only support fields in Dataset 2, the “Editorial” fields. These applications typically do not show the dataset number when referring to the field as it is assumed to be the “Editorial’ dataset.</td></tr>
<tr><td>Description</td><td>A brief description of the contents of the field.</td></tr>
</table>

Note 1:

- Standard EXIF fields are defined in “JEITA CP-3451, Exchangeable image file format for digital still cameras: Exif Version 2.2”.
- Standard TIFF fields are defined in” TIFF, Revision 6.0, Final — June 3, 1992”.
- Standard IPTC fields are defined in ” IPTC - NAA Information Interchange Model, Version No. 4, Rev 1, July 1999”

8.5.1.2 Listing Global EXIFutils Settings

The `exifinfo -g` option displays the values of global EXIFutils settings that affect the way the EXIFutils commands behave. A sample of the output from this command is shown below:

```
exifinfo -g

Global Settings
----------- Description -----------  -- Value --  -------- How Set --------
Quote Char for Command Options           '"'      exiflist /k, exifedit /c
Separator Char for Repeating Fields      ';'      exiflist /j, exifedit /j

Languages and Character Sets
------------ Usage ------------ ---- Value -----  --- Environment Variable ---
Commands Output                    OEM            EXIFUTILS_CS_CMD
Input text files                   ACP            EXIFUTILS_CS_CMD
EXIF Meta Data                     LATIN-1        EXIFUTILS_CS_EXIF
IPTC Character Set                 LATIN-1        EXIFUTILS_CS_IPTC
Store EXIF comments in UNICODE     False          EXIFUTILS_UNICODE_COMMENT
XMP Default Language               en             EXIFUTILS_XMP_LANG
```

For each setting the current value is shown, and the way in which the value is set (command options or environment variable) is shown.

8.5.1.3 Listing Known XMP Schemas

The `exifinfo -s` option lists all XMP schemas for which EXIFutils has field definitions. Sample output from this option is shown below.

```
exifinfo -s

-------------- Namespace URI --------------- ------ Description ------ - Prefix -
http://ns.adobe.com/xap/1.0/                  XMP 1.0                   xmp:
http://ns.adobe.com/xap/1.0/rights/           XMP 1.0 Rights            xmpRights:
http://ns.adobe.com/xap/1.0/mm/               XMP 1.0 Media Management  xmpMM:
http://ns.adobe.com/exif/1.0/                 XMP EXIF 1.0              exif:
http://ns.adobe.com/exif/1.0/aux/             XMP EXIF 1.0 Auxillary    aux:
http://ns.adobe.com/tiff/1.0/                 XMP TIFF 1.0              tiff:
http://ns.adobe.com/png/1.0/                  XMP PNG 1.0               png:
http://ns.adobe.com/jpeg/1.0/                 XMP JPEG 1.0              jpeg:
http://ns.adobe.com/camera-raw-settings/1.0/  XMP Camera Raw Settings   crs:
http://purl.org/dc/elements/1.1/              XMP Dublin Core           dc:
```

8.5.2 Comand Syntax

```
   exifinfo [/g|h|s|?|v]

   exifinfo /f field-list [/b]
```

where

`-b` Brief display. Use with `-f` option to display only basic field details.

`-f` [*field-name,field-name,...* | *@file-name* | *@-*]

Specifies the names of the fields that are to be displayed. This parameter can be specified in one of four ways:

1) A comma separated list of the types of metadata to be displayed. for example:

```
   -f exif,xmp
```

Value metadata types are:

- `all`   – display all metadata in the file
- `exif` – display EXIF metadata
- `iptc` – display EXIF metadata
- `xmp`   – display XMP metadata
- `rmeta` – display Ricoh RMETA metadata

2) a comma separated list of field names on the command line, for example:

```
   -f file-name,width,height,ip-caption,xex-flash.fl-fired
```

3) an "@" followed by the name of a text file containing a list of the field names to be displayed, separated by commas.

```
-f @field-list.txt
```

4) an "@" followed by a '-', indicating that the list of the field names is to be read from stdin, for example:

```
-f @-
```

`-g` List global EXIFutils settings

`-h` Display `exifinfo` usage information

`-s` Display a list of known XMP schemas.

`-v` Display program version and licensing information. This is the default action if no other options are present.

`-?` Display `exifinfo` usage information.

8.6 EXIFKEY

8.6.1 Description

The `exifkey` command installs the license key used to enable the full functionality of EXIFutils. It is also used to display a license that has already been installed.

When you purchase a registered version of EXIFutils you will be provided with a 13-digit license key. This key is generated based on the user name and email address you supplied when you purchased EXIFutils from REG.NET. This key, and the username and email address you supplied, must be installed in the EXIFutils license file on your system in order to use the full EXIFutils functionality.

If no license key is installed, or if the license key installed is invalid, the following restrictions are imposed on the operation of the EXIFutils:

- no more than 10 image files can be processed in a single invocation of any EXIFutils command,
- no more than 5 field names can be specified in a field list or field value list,
- only the first 2 fields in a template will be substituted. Subsequent fields will be replaced with the word "Demo"
- for repeating IPTC fields, only the first 2 instances of the field will be displayed. Subsequent instances will be replaced with the word "Demo".
- only the first 50 characters in IPTC field values will be displayed
- in `exifedit`, the maximum length that a string field value can be set to is 99 characters (if a valid license is installed string values can be up to 2000 characters),
- in `exifedit,` only one action parameter can be specified in a single invocation of the command, i.e. you specify one and only one of the following parameters: `-a, -e, -r, -s,` or `-t.`

To install your license key use `exifkey` as shown:

```
exifkey -k licensekey -u "username" -e emailaddr
```

where

- *`licensekey`* is the license key you received when you purchased your EXIFutils license
- *`username`* is the user name exactly as you entered it when you purchased EXIFutils
- *`emailaddr`* is the email address exactly as you entered it when you purchased EXIFutils

For example, if the username you supplied when you purchased the EXIFutils was `Jane Smith`, the email address was `jane@smith.com`, and the license key you were provide was `1234567890123`, then you would enter the `exifkey` command:

```
exifkey -k 1234567890123 -u "Jane Smith" -e jane@smith.com
```

Note that the user name and email address must be entered EXACTLY as specified when you purchased EXIFutils. If the above command correctly installs the license, then the following command would ***not*** correctly install the license:

```
exifkey -k 1234567890123 -u "jane smith" -e jane@smith.com
```

The above command would not correctly install the license because the user name has been entered as all lower case characters, whereas the first character of each name was upper case when the EXIFutils were purchased.

To display the installed license, use the `-l` option:

```
exifkey -l
```

8.6.1.1 Upgrading from EXIFutils V2.x

Upgrade license keys are available to upgrade from EXIFutils V2.x to V3. The procedure for installing an upgrade license depends on whether or not the V2.x license is already installed on the system.

If the V2.x license is already installed, then the procedure for installing the upgrade license is the same as for installing a new V3 license as described in the previous section.

If the V2.x license is NOT already installed, then install the upgrade license using the following command:

```
exifkey -k v3licensekey -u “username” -e emailaddr -o v2licensekey
```

8.6.2 Command Syntax

```
exifkey -k v3licensekey -u username -k emailaddr [-o v2licensekey]
exifkey [-h|-v|-l]
```

`-e` *emailaddr*
Email address supplied when the license key was purchased

`-h` Display `exifkey` usage information

`-k` *licensekey*
13 digit EXIFutils license key

`-l` List details of the installed license

`-o` *v2licensekey*
13 digit V2.x EXIFutils license key. Used when installing a V3 upgrade license key

`-u` *username*
User name supplied when the license key was purchased

`-v` Display program version and licensing information. This is the default action if no other options are present.

8.6.3 Examples

```
exifkey -l
```

List details of the currently installed license. Sample output is included below:

```
License Key                 : 7721245622797
Licensed User               : Jane Citizen
Licensed Email Address      : jane@citizen.com
Licensed Operating System   : Any
Installed EXIFutils Version: 3.x
License Status              : License Valid
```

If this installed license is not valid then a message similar to the following will be displayed:

```
*** License key is not correct for this user or product version. ***
*** Check that all details have been entered correctly.          ***

License Key           : 9999999999999
Licensed User Name    : Jane Citizen
Licensed Email Address: jane@citizen.com
Licensed Version      : 3.x
License Status        : Invalid
```

8.7 EXIFLIST

8.7.1 Description

The primary function of the `exiflist` command is to extract meta data field values from one or more image files, and display them in the format selected via the command parameters. The default action if no command option is specified is to display all fields.

In addition, `exiflist` is used to:

- extract a copy of the thumbnail image from the EXIF data and write it to a separate JPEG file (`-t` option). See Section 8.7.3.
- extract a copy of the audio stream from the EXIF data and write it to a separate .WAV file (`-w` option). See Section 8.7.5.
- Extract an exact copy of all Meta data (EXIF, IPTC, XMP, and Flashpix) into a separate .EXI file (`-e` option). See Section 8.7.4.

8.7.1.1 Supported Image File Types

Not all `exiflist` features can be used with all image file types. Table 28 shows the features that are supported for each image file type.

Table 28 File Types for which `exiflist` Options are Supported

File Type	No options (list all fields)	Field List (`-o` option)	EXIF Block Extraction (`-e` option)	Thumbnail Extraction (`-t` option)	Audio File Extraction (`-w` option)
Adobe .DNG	✓	✓			
JPEG	✓	✓	✓	✓	✓
TIF	✓	✓			
EXIF .EXI	✓	✓	✓	✓	
Canon .CR2	✓	✓			
Canon .CRW	✓	✓			
Fujifilm .RAF	✓	✓		✓	
Kodak .DCR	✓	✓			
Minolta .MRW	✓	✓	✓	✓	
Nikon .NEF	✓	✓			
Olympus .ORF	✓	✓			
Sony .ARW	✓	✓			
EPS	✓	✓			

8.7.2 Output Format Options

`exiflist` provides five display formats, which are selected by the `-o` option:

Format Type	Command Options	Description

Full field display	`-o d`
	`-o dn`
		Displays the values of fields in the image file in a easy to read format. The `-o d` option shows the display name of each field and the field value. The `-o dn` option shows the EXIFutils nickname for each field and the field value.
		The fields to be displayed are selected using the `-f` option.
		See Section 8.7.2.1 for more detail.

Field list	`-o l`
	`-o lh`
	`-o ln`
	`-o lf`
		Displays selected fields as a comma (or tab) separated list. The fields to be displayed are selected using the `-f` option. See Section 8.7.2.2.

SQL INSERT statement	`-o s`	Displays selected fields as an SQL INSERT statement. The fields to be displayed are selected using the `-f` option. See Section 8.7.2.3.
User specified template	`-o t`	Displays selected fields in the format specified by the user-specified *template*. The output template to be used is specified in the `-s` option. See Section 8.7.2.4.
Serialised XMP	`-o x`	Displays the serialized XMP data block. See Section 8.7.2.5.

The default action, if no formatting options are specified, is to display full details of all metadata fields (equivalent to the `-o d` option with no `-f` option specified).

8.7.2.1 Full Field Display

The full field display options display all EXIF and/or IPTC fields in an easily read format. It lists each field on its own line, with a full field name and value description. For example:

```
>exiflist -o d myphoto.jpg
FILE: myphoto.jpg

Camera
  Description               : allo
  Orientation (start point): Upper Left
  X Resolution              : 72
  Y Resolution              : 72
  Resolution Unit           : Inch
  Firmware Version          : Adobe Photoshop CS2 Macintosh
  Date/Time Modified        : 2005:09:07 15:09:51

Image
  Colour Space              : sRGB
  Image Width               : 360
  Image Height              : 216

Thumbnail
  Thumb Compression         : JPEG Compression (old style)
  Thumb X Resolution        : 72
  Thumb Y Resolution        : 72
  Thumb Resolution Unit     : Inch
  Thumb Jpeg IF Offset      : 312
  Thumb Jpeg IF Byte Count : 1725
```

```
IPTC 2: Editorial
  Dataset 2 Record Version : 2
  Object                   : Blue Square Test File - .jpg
  Keywords                 : XMP; Blue Square; test file; Photoshop; .jpg; XMP;

Photoshop
  Mark as Copyrighted      : No

XMP Dublin Core
  format                   : image/jpeg
  title                    : [Sorted List of Alternative Values]
    title[1]               :   Blue Square Test File - .jpg
      lang                 :     x-default
  rights                   : [Sorted List of Alternative Values]
    rights[1]              :
      lang                 :     x-default
    rights[2]              :
      lang                 :     en
  description              : [Sorted List of Alternative Values]
    description[1]         :
      lang                 :     x-default
    description[2]         :
```

The output from the -o dn option is similar, but shows the EXIFutils nickname of the field instead of the display name.

```
>exiflist -o d myphoto.jpg
FILE: myphoto.jpg

Camera
  description              : allo
  orient                   : Upper Left
  x-res                    : 72
  y-res                    : 72
  res-unit                 : Inch
  firm-ver                 : Adobe Photoshop CS2 Macintosh
  date-mod                 : 2005:09:07 15:09:51

Image
  colour-space             : sRGB
  width                    : 360
  height                   : 216

Thumbnail
  th-comp                  : JPEG Compression (old style)
  th-x-res                 : 72
  th-y-res                 : 72
  th-res-unit              : Inch
  th-if-offset             : 312
  th-if-count              : 1725

IPTC 2: Editorial
  ip-version               : 2
  ip-object                : Blue Square Test File - .jpg
  ip-keyword               : XMP; Blue Square; test file; Photoshop; .jpg; XMP;
```

```
Photoshop
  ip-copyrighted           : No

XMP Dublin Core
  xdc-format               : image/jpeg
  xdc-title                : [Sorted List of Alternative Values]
    xdc-title              :   Blue Square Test File - .jpg
      xdc-title            :     x-default
  xdc-rights               : [Sorted List of Alternative Values]
    xdc-rights             :
      xdc-rights           :     x-default
    xdc-rights             :
      xdc-rights           :     en
  xdc-description          : [Sorted List of Alternative Values]
    xdc-description        :
      xdc-description      :     x-default
    xdc-description        :
      xdc-description      :     en
```

8.7.2.2 Field List Display

The field list display formats are intended for use where the output from `exiflist` will be use as input to another program, eg the data is to be imported to a database or spreadsheet. These options display one line for each image file, with the fields separated by either a comma ‘,’ or a tab character. The fields to be displayed are selected using the `-f` option.

The output format to be used is selected using the sub-parameter to the `-o` option:

`-o l` **Simple Field List:** This is the simplest output format. Field values are displayed as a comma (or tab) separated value list. For example,

```
>exiflist -o l -f file-name,make,f-number myphoto.jpg
myphoto.jpg,KODAK,2.8
>
```

`-o lh` **Simple Field List with Headings:** This is similar to the `-o l` option, except that the first line lists the field nicknames of the included fields. For example,

```
>exiflist -o lh -f file-name,make,f-number myphoto.jpg
file-name,make,f-number
myphoto.jpg,KODAK,2.8
>
```

`-o lf` **List with Field Descriptions:** Each field value is preceded by its full field description followed by a colon. For example,

```
>exiflist -o lf -f file-name,make,f-number myphoto.jpg
File Name : myphoto.jpg,Make : KODAK, F-Number : 2.8
>
```

Note that no heading line is displayed.

`-o ln` **List with Nickname:** Each field value is preceded by its nickname and an equal sign. Each field value is enclosed in quotes (“). For example,

```
>exiflist -o ln -f file-name,make,f-number myphoto.jpg
file-name=”myphoto.jpg”,make=”KODAK”,f-number=”2.8”
>
```

As with the ‘`-o lf`’ option no heading line is displayed.

This output format is particularly useful as it is the same format that `exifedit -a` accepts as input. This means that `exiflist` and `exifedit` can be used in combination to transfer fields from one image file to another. For example:

```
>exiflist -o ln -f copyright photo.jpg | exifedit -a @- photo2.jpg
```

The above command copies the `copyright` field from `photo.jpg` into `photo2.jpg.`

See the `-c` option for details of selecting either commas or tab characters as the separator between fields. See the -k option for details of changing the quote character that is used.

If any of the fields values contains a new line character, it will be relaced with a semicolon character “;” so that the field output will be on one line. Use the `-n` option to specifiy an alternative character to replace newline characters.

If multiple instances of a repeating IPTC fields or XMP List fields are being listed, they will be separated by a semicolon. For example, if a file contains three instances of the IPTC keyword field with the values “trains”, “planes”, and “automobiles”, then listing the `ip-keyword` field will have the following result:

```
>exiflist -o ln -f ip-keyword photo.jpg
ip-keyword=”planes; trains; automobiles”
```

Using the `-j` option to change the character used to separate repeating IPTC fields.

8.7.2.3 SQL Display

The `-o s` option displays the fields of each image file in an SQL INSERT statement. The fields to be displayed are selected by the `-f` option. For example:

```
>exiflist -o s -f file-name,make,f-number myphoto.jpg
INSERT INTO pictures (
    file-name,
    make,
    f-number
) VALUES (
    'myphoto.jpg',
    'EASTMAN KODAK COMPANY',
    8.0
);
>
```

8.7.2.4 User Supplied Template Display

The `-o t` option displays the field values in the format described in the user supplied output template. The output template is specified in the `-s` option, which is mandatory when the `-o t` option is present. A template is a character string containing *markers* that indicate where EXIF and IPTC field values are to be substituted. A marker is any valid EXIFutils nickname surrounded by curly brackets: {}. For example:

```
>exiflist -o t -s “File: {file-name}\n  Make : {make}\n  Model: {model}\n\n” \
                                                file1.jpg file2.jpg
File: file1.jpg
  Make : NIKON
  Model: D1X

File: file2.jpg
  Make : KODAK
  Model: DC280

>
```

See Section 7 for a detailed explanation of templates.

8.7.2.5 Serialized XMP

XMP data is stored in images files in the form of a text string, in a format known as “Serialized XMP”. The `-o x` option prints the serialized XMP exactly as it appears in the file. No character conversions or formatting are applied. The Serialised XMP is stored using the UTF8 character set.

```
>exiflist -o x file1.jpg

<?xpacket begin="ï»¿" id="W5M0MpCehiHzreSzNTczkc9d"?>
<x:xmpmeta xmlns:x="adobe:ns:meta/" x:xmptk="XMP Core 4.4.0">
   <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
      <rdf:Description rdf:about=""
            xmlns:tiff="http://ns.adobe.com/tiff/1.0/">
         <tiff:Make>Canon</tiff:Make>
         <tiff:Model>Canon EOS 5D</tiff:Model>
         <tiff:Orientation>1</tiff:Orientation>
      </rdf:Description>
      <rdf:Description rdf:about=""
            xmlns:exif="http://ns.adobe.com/exif/1.0/">
         <exif:ExifVersion>0221</exif:ExifVersion>
         <exif:ExposureTime>1/200</exif:ExposureTime>
         <exif:ShutterSpeedValue>7643856/1000000</exif:ShutterSpeedValue>
         <exif:FNumber>10/1</exif:FNumber>
         <exif:ApertureValue>6643856/1000000</exif:ApertureValue>
         <exif:ExposureProgram>1</exif:ExposureProgram>
         <exif:ISOSpeedRatings>
            <rdf:Seq>
               <rdf:li>100</rdf:li>
            </rdf:Seq>
         </exif:ISOSpeedRatings>
[...]

<?xpacket end="w"?>
>
```

8.7.2.6 Selecting Fields Using the -f Option

The fields to be display are selected using `-f` option. This option can specify the fields to be displayed in two ways:

- By specifying the types of metadata to be displayed, eg: EXIF metadata, XMP metadata. See the table below for a list of valid metadata type values. These are the only values that are valid when the `-o d` or `-f dn` options are specified.

Table 29 Metadata Type Values

-f Option	**Meaning**
`-f all`	Display all metadata present in the image file.
`-f exif`	Display only EXIF metadata stored in the traditional EXIF format (i.e. EXIF fields stored in XMP format are not displayed)
`-f iptc`	Display only IPTC metadata stored in the traditional IPTC IIM format (i.e. IPTC fields stored in XMP format are not displayed)
`-f xmp`	Displays all fields stored in XMP format.
`-f rmeta`	Displays all Ricoh RMETA fields.

- By specifying a list of the fields to be displayed. For example

```
>exiflist -o ln -f file-name,make,f-number,ip-* myphoto.jpg
```

The field names can be wildcarded to select all fields whose nickname start with the same characters.

8.7.3 Thumbnail Image Extraction

Image files can contain an embedded thumbnail image, which can be extracted into a separate JPG file using the `-t` option. The `-t` option accepts as a parameter either a *`file-name-template`* that describes the name of the file into which the thumbnail is to be written, or a dot '.' indicating that the default file name template is to be used, i.e.

```
      -t file-name-template
      -t .
```

A *`file-name-template`* is a string containing markers to indicate where meta data field values, and parts of the original file name are to be inserted. See Section 7 for a detailed explanation of templates.

The file name resulting from the *`file-name-template`* may be a simple file name, or a file name with an absolute or relative directory path. If it is a relative directory path, it will be interpreted as being relative to the directory in which the image file resides, not the current working directory. For example:

```
      exiflist  -t thumbs/{file-name}  mydir/photo.jpg
```

This command will extract the thumbnail image from `mydir/photo.jpg` and store it in a file with the same name as the original image file (`photo.jpg`) in the directory `mydir/thumbs`.

The directory into which the thumbnail file will be written must already exist; `exiflist` will not create it. If the resulting file name exceeds the maximum file name size, the name will be truncated on the right. Any characters that are not valid in file names (\\/:*?\"<>|) will be replaced with a dash "-".

If no template is specified (`-t .`) then the following default template used is:

```
      {file-base}-th.jpg
```

This will create the thumbnail file in the same directory as the image file, with a name that has the characters "`-th`" inserted before the finale dot in the file name. For example:

```
      exiflist  -t .  mydir/photo.jpg
```

This command will write the thumbnail image to a file called `photo-th.jpg` in the same directory as the image file (`mydir`).

The `-t` option is not supported for all image file types. See Section 8.7.1.1 for a list of image file types that are supported by this option.

8.7.4 Meta Data Extraction

The `exiflist -e` can be used to extract a complete copy of all EXIF, XMP, IPTC, and Flashpix data from an image file into a separate `.exi` file. The .EXI file can be used to store the EXIF information while the image file is being edited, and later copied back into the image file using the `exifcopy` command. The `-e` option accepts as a parameter either a *`file-name-template`* that describes the name of the file into which the EXIF data is to be written, or a dot '.' indicating that the default template is to be used:

```
      -e file-name-template
      -e .
```

A *`file-name-template`* is a string containing markers to indicate where meta data field values, and parts of the original file name are to be inserted. See Section 7 for a detailed explanation of templates.

The file name resulting from the *`file-name-template`* may be a simple file name, or a file name with an absolute or relative directory path. If it is a relative directory path, it will be interpreted as being relative to the directory in which the image file resides, not the current working directory.
For example:

```
      exiflist  -e exifsave/{file-base}.exi  mydir/photo.jpg
```

This command will create an EXI save file with the same name as the original image file, but with the file extension changed to `.exi` (i.e. `photo.exi`) in the directory `mydir/exifsave`.

The directory into which the EXI file will be written must already exist; `exiflist` will not create it. If the resulting file name exceeds the maximum file name size, the name will be truncated on the right. Any characters that are not valid in file names (\\/:*?\"<>|) will be replaced with a dash "-".

If no template is specified (`-e .`) then the following default template used is:

```
      {file-base}.exi
```

This will create the EXIF save file in the same directory as the image file, with the file name extension changed to `.exi`. For example:

```
      exiflist  -e .  mydir/photo.jpg
```

This command will write the EXIF information to a file called `photo.exi` in the same directory as the image file (`mydir`).

The `-e` option is not supported for all image file types. See Section 8.7.1.1 for a list of image file types that are supported by this option.

8.7.5 Audio Stream Extraction

Image files can contain an embedded Flashpix audio stream, which can be extracted into a separate WAVE format file using the `-w` option. The `-w` option accepts as a parameter either a *`file-name-template`* that describes the name of the file into which the audio stream is to be written, or a dot '.' indicating that the default template is to be used:

```
      -w file-name-template
      -w .
```

A *`file-name-template`* is a string containing marks to indicate where meta data field values, and parts of the original file name are to be inserted. See Section 7 for a detailed explanation of templates.

The file name resulting from the *`file-name-template`* may be a simple file name, or a file name preceded by an absolute or relative directory path. If it is a relative directory path, it will be interpreted as being relative to the directory in which the image file resides, not the current working directory. For example:

```
      exiflist  -w audio/{file-base}-audio.wav  mydir/photo.jpg
```

This command will create a `.wav` audio file with the name `photo-audio.wav` in the directory `mydir/audio`

The directory into which the `.wav` file will be written must already exist; `exiflist` will not create it. If the resulting file name exceeds the maximum file name size, the name will be truncated on the right. Any characters that are not valid in file names (\\/:*?\"<>|) will be replaced with a dash "-".

See Section 7 for a detailed explanation of templates.

If no template is specified (`-w .`) then the default template used is:

```
      {file-base}.wav
```

This will create the WAVE audio file in the same directory as the image file, with a name that has the "`.jpg`" file extension replaced with "`.wav`". For example:

```
      exiflist  -w .  mydir/photo.jpg
```

This command will write the audio stream to a file called `photo.wav` in the same directory as the image file (`mydir`).

The `-w` option is not supported for all image file types. See Section 8.7.1.1 for a list of image file types that are supported by this option.

8.7.6 Command Syntax

```
exiflist [options] -o [d|dn|l|lh|ln|lf|s] [-f field-list] file...

exiflist [options] -o t -s output-template file...

exiflist [options] -o x file...

exiflist [options] -t [.|file-name-template] file...
exiflist [options] -w [.|file-name-template] file...

exiflist [options] -e [.|file-name-template] file...

exiflist [-h|-v]
```

where *[options]* are:

```
[/y] [/a] [-c [c|t]] [-j x] [-k x] [-n x] [/q] [-r] [-x] [/u] [/p charset-list]
```

`-a` Show absolute directory paths. When the directory name is displayed, the full directory path will be shown. By default only the path relative the current directory is shown.

`-c [c|t]` Specify the field separator character to be used by the `-l` option; `c` = comma, `t` = tab. By default commas are used.

`-e [.|`*`file-name-template`*`]`

Extract a copy of all EXIF, IPTC, XMP and Flashpix Meta data from the specified image file(s) and write it to a separate file. The directory into which the file is written, and the name of the file, is specified in the *`file-name-template`*. For example:

```
-e “{file-base}-save.{file-ext}”
```

To use the default template, specify:

```
-e .
```

The default *`file-name-template`* is:

```
{file-base}.exi
```

See Section 8.7.4 “Meta Data Extraction”, and Section 7 “Templates” or more information about the use of templates.

`-f [`*`field-name,field-name,...`*` | @`*`file-name`*` | @-]`

Specifies the names of the fields that are to be displayed by the field list options (`-o d, -o dn, -o l, -o lh, -o ln, -o lf, -o s`). This parameter can be specified in one of four ways:

5) A comma separated list of the types of metadata to be displayed. for example:

```
  -f exif,xmp
```

Value metadata types are:

- `all` – display all metadata in the file
- `exif` – display EXIF metadata
- `iptc` – display EXIF metadata
- `xmp` – display XMP metadata
- `rmeta` – display Ricoh RMETA metadata

6) a comma separated list of field names on the command line, for example:

```
  -f file-name,width,height,ip-caption
```

7) an “@” followed by the name of a text file containing a list of the field names to be displayed, separated by commas.

```
-f @field-list.txt
```

8) an “@” followed by a '-', indicating that the list of the field names is to be read from stdin, for example:

```
-f @-
```

Use the `exifinfo` command to list the valid field names that can be specified in the field list (see Section 8.5).

If the `-f` option is not specified, the following default set of fields will be displayed:

```
file-name-full
width
height
exp-time
f-number
date-taken
exp-prog
shutter
aperture
exp-bias
meter-mode
flash
```

See also Section 4.5 “Selecting Multiple Fields”.

`-h` Display `exiflist` usage information

`-j [x|0xnn|0nnn)`

Specify the field separator character to be used to separate repeating IPTC fields. This can be specified in one of three ways:

- As a printable character, for example `-j +`
- As a hexadecimal value, for example `-j 0x10`
- As an octal value with a leading zero, for example `-j 020`

The default value is a semicolon ‘`;`’

`-k x` Set the quote character used by the `-o` options to the character ‘x’. For example, to set the quote character to be a colon use:

```
-k :
```

`-n [x|0xnn|0nnn)`

Specify the character to be used to replace any new line characters that appear in field values. This can be specified in one of three ways:

- As a printable character, for example `-n +`
- As a hexadecimal value, for example `-n 0x10`
- As an octal value with a leading zero, for example `-n 020`

The default value is a semicolon ‘`;`’

`-o [d|dn|l|lh|lf|ln|s|t|x]`

Specifies the output format to be used. Available options are:

Full Field Display Options:

`-o d` Display the values of fields in the image file in an easy to read format. Shows the display name of each field and the field value.

`-o dn` Similar to `-o d` above, but displays the field nickname instead of the display name.

Field List Options:

`-o l` Display field values in a comma (or tab) separated list

`-o lh` As for `-o l` but a heading line is displayed containing field nicknames

`-o lf` Display field values preceded by their full field description

`-o ln` Display field values preceded by their field nickname. All field values are surrounded by quotes (“).

SQL Output Option:

`-o s` Display fields as an SQL INSERT statement. One INSERT statement is displayed for each file.

User-Specified Template Option:

`-o t` Display fields in the format supplied by the *`output-template`* in the `-s` option.

Serialized XMP Option:

`-o x` Display the XMP metadata in serialized XMP format.

When displaying in Full Field, List Field, or SQL format, the fields to be displayed are selected using the `-f` option. The field separator character to be used (comma or tab) is selected using the `-c` option. The separator character to be used to separate repeating IPTC fields and XMP List fields is selected by the `-j` option. The quote character to be used is selected by the `-k` option.

`-p [cmd=xxx],[exif=xxx],[iptc=xxx]`

Specify the character sets to be used when interpreting/displaying text fields

`cmd=xxx` sets the charset to be used for output displayed from this command.

`exif=xxx` sets the charset to be used for reading/writing EXIF text fields

`iptc=xxx` sets the charset to be used for reading/writing IPTC text fields

Any of the three may be omitted. For example:

```
-p exif=latin1,iptc=mac
-p cmd=utf8
```

These values can also be set using the `EXIFUTILS_CS_CMD`, `EXIFUTILS_CS_EXIF`, `EXIFUTILS_CS_IPTC` environment variables.

See Section 6 “Character Sets” for more information.

`-q` Quiet mode. Do not display warning messages

`-r` Include subdirectories. If any of the files specified is a directory, then image files in the specified directory and all of its subdirectories will be processed.

`-s [`*`template`*`|`*`@file-name`*`|@-]`

Specifies the *`output-template`* to be used by the `-o t` option. This option can be specified in one of three ways:

1) on the command line, for example:

```
-s “{file-name}\t{date-taken}\t{model}”
```

2) as an “@” followed by the name of a text file containing the *`output-template`*:

```
-s @template.txt
```

3) as an “@” followed by a '-', indicating that the *`output-template`* is to be read from stdin, for example:

```
-s @-
```

`-t [.|`*`file-name-template`*`]`

Extract a copy of the thumbnail image from the specified image file(s) and write them to a separate file. The directory into which the file is written, and the name of the file, is specified in the *`file-name-template`*. For example:

```
-t “{file-base}-thumb.{file-ext}”
```

To use the default thumbnail template, specify:

```
-t .
```

The default `file-name-template` is:

```
{file-base}-th.jpg
```

See Section 8.7.3 “Thumbnail Image Extraction”, and Section 7 for more information about the use of templates.

`-u` Unformatted mode. `exiflist` will not interpret field values, it will display them as raw values. For example, the shutter speed value `1/125` sec will be displayed uninterpretted as `70/10` in unformatted mode.

`-v` Display program version and licensing information. This is the default action if no other options are present.

`-w [.|`*`file-name-template`*`]`

Extract a copy of the Flashpix audio stream from the specified image file(s) and write it to a separate file. The directory into which the file is written, and the name of the file, is specified in the *`file-name-template`*. For example:

```
-w “{file-base}-audio.WAV”
```

To use the default audio file name template, specify:

```
-w .
```

The default `file-name-template` is:

```
{file-base}.wav
```

See Section 8.7.5 “Audio Stream Extraction” and Section 7 “Templates” for more information about the use of templates.

`-x` Microsoft Excel compatibility mode. When importing `exiflist` output into Microsoft Excel, Excel does not correctly interpret some EXIF field values (eg: Microsoft Excel™ does not recognise the EXIF date/time format). In Microsoft Excel compatibility mode `exiflist` adjusts the format of certain fields so that Microsoft Excel interprets them correctly. This option is ignored if unformatted mode (`-u`) is also specified.

`-y` Used when invoking this command from a .BAT file that has been created with an ANSI text editor. This option translates the command parameters so that non-English characters are interpretted correctly. If present this option MUST be the first option on the command.

`-?` Display `exiflist` usage information.

8.7.7 Examples

```
exiflist image1.jpg image2.tif
```

Displays all EXIF, IPTC, and RMETA information from the image files `image1.jpg` and `image2.tif`

```
exiflist -q -r -o l .
```

Displays a summary of EXIF information from all image files in the current directory and all of its subdirectories (`-r` option). Information will be displayed one line per image file (`-o l` option), and warning messages will be suppressed (`-q` option).

```
exiflist -o l -c t ../pics /usr/users/fred image1.jpg
```

Displays EXIF and IPTC information from all image files in the two specified directories (`../pics` and `/usr/users/fred`), and for the file `image1.jpg` in the current directory. Information will be displayed one line per file (`-o l` option) with a tab character separating each field (`-c t` option). As the `-f` option is not present, a selection of commonly used EXIF fields will be displayed (see Section 4.5 for more information on the commonly used fields that will be displayed).

```
exiflist -o lh -f file-name,width,height .
```

Displays the file name, image width, and image height, one line per image file, for all image files in the current directory. One line will be printed for each file. A heading line listing the field names will be displayed as the first line (`-o lh` option).

```
exiflist -o lh -f @file-list.txt .
```

For all image files in the current directory, displays the fields specified in the file `file-list.txt`. A heading line listing the field names will be displayed as the first line (`-o lh` option). The contents of `file-list.txt` must be a comma-separated list of field nicknames, for example:

```
description, comment,
date-taken, f-number,shutter
```

New line characters in the file are ignored by `exiflist`.

```
exiflist -o l -f aperture image.jpg
```

Displays the camera aperture setting used when `image.jpg` was taken.

```
exiflist -l gps\*,width,height
```

List the descriptions and valid values for all fields whose nickname begins with “`gps`”, and for the `width` and `height` fields.

```
exiflist -o t -s @template.txt mydir
```

Displays fields from all image files in `mydir`, using the output format specified in the output-template read from `template.txt`. A sample `template.txt` is included below:

```
File-name: {file-name}
      Taken on: {date-taken}
      Shutter Speed: {shutter}
      F-Stop: {f-number}
      Compensation: {exp-bias}
```

With this template, the output would be as follows:

```
File-name: myphoto1.jpg
      Taken on: 2003:01:01 07:00:00
      Shutter Speed: 1/125
      F-Stop: 2.8
      Compensation: 0.00

File-name: photo2.jpg
      Taken on: 2003:01:01 08:14:34
      Shutter Speed: 1/60
      F-Stop: 5.6
      Compensation: -0.5
```

```
exiflist -t “{file-base}-thumb.jpg” image1.jpg
```

Extracts the EXIF thumbnail file from the image file `image1.jpg` and writes it to a file called `image1-thumb.jpg`.

```
exiflist -e . image1.jpg
```

Takes a copy of the Meta data from the file `image1.jpg` and writes it to a file called `image1.exi.`

```
exiflist -o i -p iptc=mac image1.jpg
```

Display all IPTC fields that are present in image1.jpg. All IPTC text fields in the image will be interpreted using the Mac Roman character set.

9 Software License Agreement

Carefully read this license agreement before using the product. Using the product indicates that you have read this license agreement and agree to its terms.

By using the product you accept the following License Agreement. If you do not agree to the conditions of this license, delete the product from your system.

Unregistered User License: The Unregistered User License entitles you to use a restricted set of product functionality for an unlimited time on any single computer without purchasing a license key. The software is in use when it is either loaded into RAM or installed into the hard disk or other permanent memory. The product functionality provided by the Unregistered User license is described in the Users Guide.

Registered User License: To use the full functionality of the product a Registered User license must be purchased. Installing the Registered User license key enables the full functionality of the product for an unlimited time. A Registered User license permits you to use one copy of the software included in this package on any single computer. For each software license, the program can be in use on only one computer at any given time. The software is in use when it is either loaded into RAM or installed into the hard disk or other permanent memory.

The software is protected by United States copyright laws, international treaty provisions, and all other applicable national laws. The software must be treated like all other copyrighted materials (e.g. books and musical recordings).

OTHER RESTRICTIONS. This License Agreement is your proof of license to exercise the rights granted herein and must be retained by you. LICENSEE may not rent or lease the Product. LICENSEE may not reverse engineer, decompile or disassemble the software program except to the extent such foregoing restriction is expressly prohibited by applicable law.

NO WARRANTY. ANY USE BY LICENSEE IS AT THE LICENSEE'S OWN RISK. PRODUCT IS PROVIDED FOR USE "AS IS" WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED BY LAW, THE AUTHOR DISCLAIMS ALL WARRANTIES OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. THE AUTHOR IS NOT OBLIGATED TO PROVIDE ANY UPDATES TO THE PRODUCT.

NO LIABILITY FOR CONSEQUENTIAL DAMAGES. In no event shall the author be liable for any damages whatsoever (including, without limitation, incidental, direct, indirect and consequential damages, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use or inability to use this product, even if advised of the possibility of such damages. Because some states/countries do not allow the exclusion or limitation of liability for consequential or incidental damages, the above limitation may not apply to you.

This shareware software may be freely distributed, provided that; (1) Such distribution includes only the original archive. You may not alter, delete or add any files in the distribution archive. (2) The distribution does not include a registration key. (3) No money is charged to the person receiving the software, beyond reasonable cost of packaging and other overhead.

10 Version History

See Section 1.5 “What’s New in V3.0?” for features and bug fixes in this release.

11 Index