Totally Objects EXIF Reader

15 June 2022

Introduction
Buried within each and every JPEG is a whole string of data about the image. This is known as EXIF and comprises a list of "tags". EXIF tags are stored as byte streams within the JPEG file and each has a different makeup from all the others.

This code has been developed to enable a user of Instantiations VAST Smalltalk to interrogate any JPEG and extract the EXIF data held within. It is a standalone application that requires no prerequisites. There is a small list of class methods that provide the results covering the main requirements.

EXIF data
Each EXIF tag definition has the same format as all others as follows:

> Tag (Hex)
> Type (Byte, ASCII, Short, Long, Rational, Undefined, Long, Rational)
> Count (Number of values - see later)
> Default (The initialised value).

EXIF data is defined by the Japanese Electronics and Information Technology Industries Association (JEITA). The software has been developed using their document JEITA CP-3451 Exchangeable Image file format for digital still cameras: Exif Version 2.2. The latest version CP-3451C (2.3 dated April 2010) is available from https://rb.gy/zphlvn.

VAST data object
tagName - the output name of the tag
fieldName - the EXIF defined name of the tag
count - as above
tagDec - decimal representation of the Tag as above
type - as above
originalNames - Type as above (an array)
typeNames - originalNames as Smalltalk symbols
tagHex - Tag as above

Usage

There are four class methods provided that give access to the EXIF data within a single JPEG.

SoSExifReader createCompleteTagListForFile: aFilePath
SoSExifReader createTagListForFile: aFilePath usIng: aTagList
SoSExifReader parseAFileCalled: aFileName
SoSExifReader showTagListFor: aJPEGFile

There is one further class method that shows the generic list of Tags

SoSExifReader showTagDefinitions

The image is passed as a file name in the form "test image.jpg'. No path is required for images in the Smalltalk root directory - normally CfsDirectoryDescriptor startUpDirectoryPath

> ***SoSExifReader createCompleteTagListForFile: aFilePath***
> Creates and answers complete tag list with data for the chosen image
>
> ***SoSExifReader createTagListForFile: aFilePath usIng: aTagList***
> Creates complete tag dictionary with data for the chosen image and returns only the tags in aTagList. Data returned is the actual data for each entry in the dictionary. The tag names in the list should be
>
> ***SoSExifReader parseAFileCalled: aFileName***
> An Execute will show all the tags for the current image to the Transcript. an Inspect will also answer a complete list of the EXIF tags with their constituent data (see EXIF Data above).
>
> ***SoSExifReader showTagListFor: aJPEGFile***
> This provides an OrderedCollection of the available tags within the chosen image.
>
> ***SoSExifReader showTagDefinitions***
> Answers a Dictionary of all EXIF tags with their constituent data.

Requirements

There is one requirement for the program to execute correctly. It requires a list of tag data to be loaded before it can analyse any image. This data is provided within a CSV file called tags.csv. This file MUST be within the root directory of the calling program. The data held within this CSV is provided in the appendix of this document.

Testing
Each of the class methods has a comment that includes the code to execute with a sample JPEG. This JPEG is called test image.jpg and should be in the root directory of the image. If it is elsewhere then insert a path before execution.

In Use
The above mentioned class methods are the ones to call to provide the required data. When providing a tag list, this should be a list of the #fieldname for each tag - see the example in the appropriate

comment.

Installation
The program is provided as a VAST application and should be loaded using the Applications Editions menu option and then the Import menu item. Once installed, it needs to be loaded. Tags.CSV and test image.jpg should be placed in the root directory and then all the class methods in SoSEXIFReader can be executed.

Warranty
This program is provided without support. We cannot guarantee that we will resolve any issues you may find. Paid support on an hourly or contract basis is available.

Usage
The software is free for use but is supplied without responsibility for its use or results. Any use should rigorously test the software before using.

Totally Objects