Exiv2 - Feature #465
Support standalone JPEG APP1 segments as an Image format

10 Apr 2006 19:53 - Andreas Huggel

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Robin Mills	**% Done:**	100%
Category:	not-a-bug	**Estimated time:**	0.00 hour
Target version:	0.26		

Description

JPEG parser should be able to read/write standalone JPEG APP1 segments, leave the code common in JpegImage.
An App1Image class derived from JpegBase will deal with the format recognition.

Additional information:

From an email exchange with Gilles Caulier:

This seems to be a complete JPEG APP1 Exif segment. To support this in Exiv2,
I'll need to tweak the JPEG parser a bit to also accept such a segment alone,
but will have to be very careful to avoid side effects.
A quick hack is faster: The TIFF header starts 10 bytes into the file, so just
add 10 from the beginning of the data buffer and use ExifData::load...

```
    // Read the file into a memory buffer
        long len = io.size();
        DataBuf buf(len);
        io.read(buf.pData_, buf.size_);
        if (io.error() || io.eof()) throw Error(14);
    ExifData exifData;
        int rc = exifData.load(buf.pData_ + 10, buf.size_ - 10);
```

-ahu.

On Tuesday 11 April 2006 04:44, you wrote:

> *Andreas,*
>
> *Here we are a bin raw exif file extrcated from my Minolta Dynax 5D camera*
> *using gphoto2 interface (via digikam camera gui)*
>
> *This file cannot be parsed by Exiv2. I don't know why...*
>
> *But, this file can be parsed using libexif (via old implementation of*
> *digikam).*
>
> *Can you give me your viewpoint ? This problem can be fixed easily into*
> *Exiv2 core ?*
>
> *Thanks in advance*
>
> *Gilles Caulier*

Related issues:

Related to Exiv2 - Feature # 467: Interface to access (Exif) metadata in bina...	**New**

History

#1 - 04 May 2010 13:45 - Jim Nelson

Just a note: the fix listed above is out-of-date. I had to use this to work around the problem:

Exiv2::Image::AutoPtr image = Exiv2::ImageFactory::create(Exiv2::ImageType::jpeg);
Exiv2::ExifParser::decode(image->exifData(), data + 10, n_data - 10);

#2 - 22 Aug 2015 09:39 - Robin Mills

- Description updated

- Category changed from metadata to not-a-bug

- Status changed from New to Closed

- Assignee set to Robin Mills

- Target version set to 0.26

I don't believe there is such a file format. It is not listed here: https://en.wikipedia.org/wiki/List_of_file_signatures

It would be possible to write a little filter *app1tojpeg* to wrap such the file into a genuine JPEG which can be handled by Exiv2.

#3 - 22 Aug 2015 16:16 - Robin Mills

- % Done changed from 0 to 100

Files

exif_from_gphoto_camera.dat	52.4 KB	10 Apr 2006	Redmine Admin