

Exiv2 - Bug #1113

Crash in Exiv2 0.25

31 Aug 2015 16:56 - Harry McKame

Status:	Closed	Start date:	31 Aug 2015
Priority:	Normal	Due date:	
Assignee:	Robin Mills	% Done:	100%
Category:	basicio	Estimated time:	4.00 hours
Target version:	0.26		
Description			
<p>I have succeeded in creating a test case for the crash, which I enclose here. The problem is memory overrun, but I cannot debug it very much for some unknown reason.</p> <p>The crash message is :</p> <pre>Unhandled exception at 0x761dc42d in test.exe: Microsoft C++ exception: Exiv2::BasicError<char>; at memory location 0x0leef0c0.</pre> <p>The enclosed file Eiv2demo.7z was created from the folder C:\Eiv2demo, designed to be stand-alone in the sense of containing as subfolders the exiv2 include, lib and dll files; something you might prefer to change.</p> <p>The folder C:\Eiv2demo is the test folder where are stored all executables and data. Its sub-folders are :</p> <ul style="list-style-type: none">- ReleaseDLL, DebugDLL : the exiv2 .lib files- Include : the exiv2 .h files- Test : The .sln that creates test.exe in C:\Eiv2demo- Interdll : The .sln that creates the interface interdll.dll in C:\Eiv2demo <p>This DLL does the calls to exiv2.</p> <ul style="list-style-type: none">- root folder C:\Eiv2demo : contains test.exe, interdll.dll, the exiv2 dlls and the test image <p>What the test program does (as approximation of our product) :</p> <ol style="list-style-type: none">1. copy the image file 321786_2.jpg to test.jpg (for repeated testing)2. Starts a thread which does the test3. The thread reads all metadata tags from test.jpg, keeping them in memory, except for one tag whose value it changes4. The thread deletes all metadata in test.jpg5. The thread re-creates all the stored tags in global variables containing exiv2 tag-repositories6. The thread assigns these repositories to the image7. The metadata is written to test.jpg (crash happens in the previous step) <p>To see the crash before you start debugging, just execute the furnished test.exe. You might like to reset in test.sln and interdll.sln the include and lib folders to yours.</p> <p>Remark : I have used Visual Studio 2008. If you wish to convert them to VS2005, perhaps this will help : http://stackoverflow.com/a/58286/165358</p> <p>Side question: How do I get this website to inform me about changes only to my questions? I tried to set it to Watch, but now I get all messages of everybody, which is quite confusing, and Unwatch doesn't seem to help.</p> <p>Thanks in advance.</p>			
Related issues:			
Related to Exiv2 - Feature #640: method is missing in Exiv2 to get list of kn...		Closed	18 Jul 2009
Related to Exiv2 - Bug #751: adobe xmp namespace		Closed	17 Jan 2011

History

#1 - 31 Aug 2015 17:08 - Robin Mills

- Category set to basicio
- Status changed from New to Assigned

- Assignee set to Robin Mills
- Priority changed from High to Normal
- Target version set to 0.26

Harry

I don't know the answer your question about when Redmine sends email. I believe "Watch" will subscribe you to individual issues and forum topics.

Can I ask you to simplify this test case. Whenever possible, it's much easier for me to reproduce your situation if you can demonstrate it with exiv2(.exe) or another sample application. And if that's not possible, maybe you can modify samples/exifprint.cpp and let me have your "crasher" code - that's so much quicker and easier for me.

I have all the major editions of Microsoft Visual Studio (2005/8/10 and 12). Our build server uses 2005 by default.

#2 - 31 Aug 2015 17:13 - Harry McKame

I'll try to simplify it tomorrow as much as I can.

#3 - 31 Aug 2015 20:59 - Robin Mills

Can you check that you are linking debug DLLs everywhere, or release DLLs. Mixing debug/release DLLs can lead to problems: <http://dev.exiv2.org/boards/3/topics/2192>

#4 - 01 Sep 2015 08:48 - Harry McKame

- File Eiv2demo.7z added

Mixing: I used my own compiled debug DLLs from ..msvc2005\bin\Win32\DebugDLL. I don't believe there was any mixing, but all is possible (Murphy).

I have managed to simplify the test case a lot. It is now one exe without threads, and all it does is :

1. store tags in 3 global exiv2 tag-repositories (for iptc, exif and xmp)
2. assign these 3 repositories to the image test.jpg (crash happens when assigning xmp)
3. write the metadata to test.jpg

The attached file contains my zipped folder "C:\Eiv2demo".
The "Test" sub-folder contains the test solution files, test.exe is linked into C:\Eiv2demo.
"DebugDLL" contain the exiv2 debug .lib files.
"Include" contains the exiv2 header files.

#5 - 01 Sep 2015 10:42 - Robin Mills

Thanks, Harry. I'll investigate this evening (I'm in England). We'll get this fixed today. I'm 100% confident.

Just taking a quick "peek" at your code with the editor, I have a few thoughts:

- 1) I'm puzzled. Is there code to initialize list missing? Possibly a loop over argv or something?

```
for (p1list = list; p1list->ptag != 0; p1list++)
```

- 2) The code in SetComment() looks safe enough on quick inspection.

- 3) Is it possible to simplify this into a single file and forget about the DLL. Is there any reason not to put the code from interdll.cpp into test.cpp. Once we know the code's solid, we can hive it off into a DLL.

- 4) Put a try { ... } catch (Exiv2::Error& e) { ... } round main() to get more information about the exception. Just like in samples/exifprint.cpp:

```
int main(int argc, char* const argv[])
try {
    if (argc != 2) {
        std::cout << "Usage: " << argv[0] << " file\n";
        return 1;
    }
    ... deleted ...
    return 0;
}
catch (Exiv2::Error& e) {
    std::cout << "Caught Exiv2 exception '" << e.what() << "'\n";
```

```
    return -1;
}
```

#6 - 01 Sep 2015 11:21 - Harry McKame

Hi Robin,

1) "list" is a memory structure, whose data is found in the file struct.h, and which is initialized in test.cpp via :
#include "struct.h"

It's an array of struct containing a pair of pointers : { ptag , pvalue }.
There are about 171 such pairs in the array, terminated by a null entry.

2) I have been through that code 1000 times, and it all looks correct.

3) Already done - no more dll.

test.cpp is a trivial program that calls functions in interdll.cpp (the "dll" part is of archaeological interest only).
All is in one simple exe.

4) try-catch will not give a better error message, since this error is not thrown by the exiv2 software.
It's a hard memory exception (I would guess because of using a zero pointer), so can only be caught using a debugger.

If I had to guess, I would say that the xmp code conflicts with iptc/exif, since
if I omit the meta-tags with "Iptc.*" and "Exif.*", leaving only "Xmp.*", there is no crash.

#7 - 01 Sep 2015 17:21 - Robin Mills

Fixed. I added the try .. catch guard and the message was:

```
Caught Exiv2 exception 'No namespace info available for XMP prefix `illustrator'`
Press any key to continue . . .
```

So I've added calls to registerNs and your code works fine. Here's the output:

```
506 rmills@rmillsmbp-w7:/cygdrive/y/Downloads $ exiv2 -pa -g illustrator /cygdrive/c/Eiv2demo/test.jpg
Warning: Directory Thumbnail, entry 0x0201: Data area exceeds data buffer, ignoring it.
Xmp.illustrator.StartupProfile          XmpText      5  Print
507 rmills@rmillsmbp-w7:/cygdrive/y/Downloads $
```

And here's the code:

```
// test.cpp : Defines the entry point for the console application.
//

#ifdef _WIN32_WINNT
#define _WIN32_WINNT 0x0600
#endif

#include <windows.h>

#include <process.h>
#include <stdio.h>
#include <tchar.h>

#include <exiv2/exiv2.hpp>

typedef struct tllist {
    char* ptag;
    char* pvalue;
} tllist;
#include "struct.h"

int    errflag = -1;          /* flag to stop */

extern void EraseComments();
extern int  SetComment(const char * ptag, const char * pvalue);
extern int  PutImageMetaFile(const char * pfilename, char ** perrmsg);

int main(int argc, _TCHAR* argv[])
try {
    tllist *pplist;
    char *perrmsg;

    Exiv2::XmpProperties::registerNs("http://clanmills.com/illustrator","illustrator");
    Exiv2::XmpProperties::registerNs("http://clanmills.com/fwc","fwc");
```

```

Exiv2::XmpProperties::registerNs("http://clanmills.com/fwr","fwr");

EraseComments();
for (pplist = list; pplist->ptag != 0; pplist++)
    SetComment(pplist->ptag, pplist->pvalue);
errflag = PutImageMetaFile("C:\\Eiv2demo\\test.jpg", &perrmsg);
return(0);
} catch (Exiv2::Error& e) {
    std::cout << "Caught Exiv2 exception '" << e.what() << "'\n";
    return -1;
}
}

```

Some other points:

1. You'll have to modify the include path to exiv2-directory/include to find "<exiv2/exiv2.hpp>"
2. don't use <http://clanmills.com/anything> as a namespace URI - get the correct URI from Adobe (and fwc/fwr)
3. The message about 0x201 is usually harmless and caused by incorrect exif data references in the file

There are more than 60 registered namespaces in libexiv2 and xmppsdk. You can list them from the command line with

```
exiv2 --verbose --version -g xmlns
```

--verbose --version -g grep-pattern has been in Exiv2 for a couple of years, however I only added reporting xmlns a couple of weeks ago.

It's OK to call registerNs on a preregistered NS provided you use the same prefix and URI. Better to avoid re-registering namespaces. We have an API in development for v0.26 for you to obtain a map of the registered namespaces. [#640](#)

#8 - 01 Sep 2015 18:47 - Harry McKame

Robin,

Thanks very much, but your solution seems to me like highly concentrated black magic.

My problems are:

1. The exiv2 command you gave produces only the version number as output. Without "-g" I get what looks like build options but no namespaces.
 2. I can't find any documentation or guidelines for using registerNs with xmp
 3. Why do you registerNs a URL from your family's website? I don't think I should do the same. And in addition they do not exist.
- Specifying dummy URLs to make xmp tags work looks really weird.
4. I suppose that the "illustrator" problem is caused by the tag "Xmp.illustrator.StartupProfile", but how come exiv2 is capable of passing me tags it cannot receive back ?
 5. How can I avoid the need for specifying namespaces, or what to specify ? (quote from above : "more than 60 registered namespaces in libexiv2 and xmppsdk" !!)
 6. If I need to specify URLs for xmp to work, what do I do when the Internet is not available ?

Any documentation that explains the problem and its solution would be more than appreciated.

#9 - 01 Sep 2015 19:20 - Alan Pater

Harry McKame wrote:

2. I can't find any documentation or guidelines for using registerNs with xmp

http://www.exiv2.org/doc/xmpsample_8cpp-example.html

```

// -----
// Register a namespace which Exiv2 doesn't know yet. This is only needed
// when properties are added manually. If the XMP metadata is read from an
// image, namespaces are decoded and registered at the same time.
Exiv2::XmpProperties::registerNs("myNamespace/", "ns");

```

Command line is on this example page: www.exiv2.org/sample.html

```

# Register a namespace which Exiv2 doesn't know yet with a prefix.
reg ns myNamespace/

```

3. Why do you registerNs a URL from your family's website? I don't think I should do the same. And in addition they do not exist. Specifying dummy URLs to make xmp tags work looks really weird.

Exactly. As Robin said, don't use those dummy sample namespaces, look up the correct ones.

They are:

```
illustrator = "http://ns.adobe.com/illustrator/1.0/"  
fwc = "http://ns.fotoware.com/iptcxmp-custom/1.0/"  
fwr = "http://ns.fotoware.com/iptcxmp-reserved/1.0/"
```

6. If I need to specify URLs for xmp to work, what do I do when the Internet is not available ?

While XMP namespaces look like URL's, they are actually just a text string. Applications do not actually try to connect to them as URL's, nor is there any requirement for them to actually exist on the Internet. No connectivity is needed.

XMP stands for "Extensible Metadata Platform". That means it can be and has been extended. Everyone and her sister can and has created their own custom namespace. exiv2 cannot possibly include every obscure namespace on the planet. That is why it has the function to register any unknown namespace on the fly. If you can make a business case for including these namespaces and provide a link to the schema specification, I would be happy to add them to exiv2.

The XMP specification can be found here: <http://www.adobe.com/devnet/xmp.html>

#10 - 01 Sep 2015 20:29 - Harry McKame

Alan,

Thanks for your answer. I apologize for still being mystified.

You say :

"While XMP namespaces look like URL's, they are actually just a text string. Applications do not actually try to connect to them as URL's, nor is there any requirement for them to actually exist on the Internet. No connectivity is needed."

So why does exiv2 crash in weird ways if these useless strings are not specified?
(small remark: It really should terminate more gracefully. Putting every call in try-catch is cumbersome.)

I have written XMP-processing software, and I understand that knowledge of the schema is required to process the tags.
It is still strange to me that by specifying a non-existent schema Robin managed to get it working.
I will see what comes out of his code, but a correct XMP packet needs its RDF.

This raises new questions:

1. If the namespaces are never actually read, where does the RDF come from when adding XMP to an image that doesn't have it?
How can exiv2 work correctly without a schema unless it's built-in (and which schemata are built-in) ?
2. How can I detect which XMP tag is problematic before adding it to Exiv2::XmpData ?
3. Do I always need to call readMetadata() before calling writeMetadata() ?

#11 - 01 Sep 2015 20:34 - Robin Mills

Thanks to Alan for answering most of Harry's questions. It only looks like magic. I assure you that it isn't. And please believe I have not attempted to confuse you.

1. The exiv2 command you gave produces only the version number as output.
Without "-g" I get what looks like build options but no namespaces.

I only added the xmlns output recently to exiv2 --verbose --version ([r3891](#)). I'll post the output from the current trunk ([r3916](#)) below.

2. I can't find any documentation or guidelines for using registerNs with xmp

I think Alan has answered this.

3. Why do you registerNs a URL from your family's website? I don't think I should do the same.
And in addition they do not exist.
Specifying dummy URLs to make xmp tags work looks really weird.

It's not a URL, its a URI. URL = uniform resource **location** URI = uniform resource **identifier**.
https://en.wikipedia.org/wiki/Uniform_resource_identifier

This is "bread-and-butter" XML. I made up something unique to get your code to work, and, although I know about illustrator, I didn't know fwc and fwr. I wanted to alert you to this and instead confused you. Prefixes such as "illustrator" have a standard definition defined by Adobe. You should always use the standard definition.

4. I suppose that the "illustrator" problem is caused by the tag "Xmp.illustrator.StartupProfile", but how come exiv2 is capable of passing me tags it cannot receive back ?

If you have a file that has embedded XMP, it can contain XML such as:

```
<xmlns:illustrator="http://...adobe...">
```

The <xmlns:prefix="URI"> syntax is standard XML. When the XMP is read (by the Adobe XMPsdk which is compiled into libexiv2), it will register the illustrator namespace dynamically. Once registered, XMPsdk (and so libexiv2) respects that namespace. So, although "illustrator" isn't in the pre-registered namespaces, exiv2 can emit Xmp.illustrator keys.

I suspect you obtained the code in your file struct.h from output from exiv2.exe -pa one-of-your-test files.jpg. Indeed, that file *one-of-your-test files.jpg* will contain <xmlns:illustrator="http://...adobe...">. To push the Xmp.illustrator.something key into your file test.jpg, you need to register the namespace. This isn't a "strange little way of libexiv2", its XML. You must register a namespace to use it.

5. How can I avoid the need for specifying namespaces, or what to specify ?
(quote from above : "more than 60 registered namespaces in libexiv2 and xmpsdk" !!)

The point I'm making is that we have quite a number of namespaces built into libexiv2 which you can use. For example:

```
955 rmills@rmillsmbp:~/gnu/exiv2/trunk $ exiv2 -vV -g microsoft
exiv2 0.25 001900 (64 bit build)
xmlns=MP:http://ns.microsoft.com/photo/1.2/
xmlns=MPRI:http://ns.microsoft.com/photo/1.2/t/RegionInfo#
xmlns=MPReg:http://ns.microsoft.com/photo/1.2/t/Region#
xmlns=MicrosoftPhoto:http://ns.microsoft.com/photo/1.0/
xmlns=expressionmedia:http://ns.microsoft.com/expressionmedia/1.0/
956 rmills@rmillsmbp:~/gnu/exiv2/trunk $
```

We have photoshop:

```
956 rmills@rmillsmbp:~/gnu/exiv2/trunk $ exiv2 -vV -g photoshop
exiv2 0.25 001900 (64 bit build)
xmlns=photoshop:http://ns.adobe.com/photoshop/1.0/
957 rmills@rmillsmbp:~/gnu/exiv2/trunk $
```

Surprisingly, we don't have illustrator. Why not? Nobody knows. I doubt if my buddies at Adobe can explain that either. Remember, we're building libexiv2 above Adobe XMPsdk and Adobe didn't include illustrator.

6. If I need to specify URLs for xmp to work, what do I do when the Internet is not available ?

The namespace prefix/URI is standard XML syntax. It doesn't require a connection to the internet.

One final point. This might seem like some kind of magic. It's supported magic. Alan and I and other members of Team Exiv2 are here and willing to help you learn about metadata and libexiv2. We didn't invent any of this. We are confident that our technology and support will help you achieve your goals.

Here's the promised output from r3916:

```
958 rmills@rmillsmbp:~/gnu/exiv2/trunk $ exiv2 -vV -g xmlns -g svn
exiv2 0.25 001900 (64 bit build)
svn=3916
xmlns=DICOM:http://ns.adobe.com/DICOM/
xmlns=GPano:http://ns.google.com/photos/1.0/panorama/
xmlns=Iptc4xmpCore:http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/
xmlns=MP:http://ns.microsoft.com/photo/1.2/
xmlns=MPRI:http://ns.microsoft.com/photo/1.2/t/RegionInfo#
xmlns=MPReg:http://ns.microsoft.com/photo/1.2/t/Region#
xmlns=MicrosoftPhoto:http://ns.microsoft.com/photo/1.0/
xmlns=acdsee:http://ns.acdsee.com/iptc/1.0/
xmlns=album:http://ns.adobe.com/album/1.0/
xmlns=asf:http://ns.adobe.com/asf/1.0/
xmlns=aux:http://ns.adobe.com/exif/1.0/aux/
xmlns=bmsp:http://ns.adobe.com/StockPhoto/1.0/
xmlns=creatorAtom:http://ns.adobe.com/creatorAtom/1.0/
xmlns=crs:http://ns.adobe.com/camera-raw-settings/1.0/
xmlns=dc:http://purl.org/dc/elements/1.1/
xmlns=dcterms:http://purl.org/dc/terms/
xmlns=digiKam:http://www.digikam.org/ns/1.0/
xmlns=dwc:http://rs.tdwg.org/dwc/index.htm
xmlns=exif:http://ns.adobe.com/exif/1.0/
xmlns=exifEX:http://cipa.jp/exif/1.0/
xmlns=expressionmedia:http://ns.microsoft.com/expressionmedia/1.0/
```

```

xmlns=iX:http://ns.adobe.com/iX/1.0/
xmlns=iptcExt:http://iptc.org/std/Iptc4xmpExt/2008-02-29/
xmlns=jp2k:http://ns.adobe.com/jp2k/1.0/
xmlns=jpeg:http://ns.adobe.com/jpeg/1.0/
xmlns=kipi:http://www.digikam.org/ns/kipi/1.0/
xmlns=lr:http://ns.adobe.com/lightroom/1.0/
xmlns=mediapro:http://ns.iview-multimedia.com/mediapro/1.0/
xmlns=mwg-kw:http://www.metadataworkinggroup.com/schemas/keywords/
xmlns=mwg-rs:http://www.metadataworkinggroup.com/schemas/regions/
xmlns=pdf:http://ns.adobe.com/pdf/1.3/
xmlns=pdfaExtension:http://www.aiim.org/pdfa/ns/extension/
xmlns=pdfaField:http://www.aiim.org/pdfa/ns/field#
xmlns=pdfaProperty:http://www.aiim.org/pdfa/ns/property#
xmlns=pdfaSchema:http://www.aiim.org/pdfa/ns/schema#
xmlns=pdfaType:http://www.aiim.org/pdfa/ns/type#
xmlns=pdfaid:http://www.aiim.org/pdfa/ns/id/
xmlns=pdfx:http://ns.adobe.com/pdfx/1.3/
xmlns=pdfxid:http://www.npes.org/pdfx/ns/id/
xmlns=photoshop:http://ns.adobe.com/photoshop/1.0/
xmlns=plus:http://ns.useplus.org/ldf/xmp/1.0/
xmlns=png:http://ns.adobe.com/png/1.0/
xmlns=rdf:http://www.w3.org/1999/02/22-rdf-syntax-ns#
xmlns=stArea:http://ns.adobe.com/xmp/sType/Area#
xmlns=stDim:http://ns.adobe.com/xap/1.0/sType/Dimensions#
xmlns=stEvt:http://ns.adobe.com/xap/1.0/sType/ResourceEvent#
xmlns=stFnt:http://ns.adobe.com/xap/1.0/sType/Font#
xmlns=stJob:http://ns.adobe.com/xap/1.0/sType/Job#
xmlns=stMfs:http://ns.adobe.com/xap/1.0/sType/ManifestItem#
xmlns=stRef:http://ns.adobe.com/xap/1.0/sType/ResourceRef#
xmlns=stVer:http://ns.adobe.com/xap/1.0/sType/Version#
xmlns=tiff:http://ns.adobe.com/tiff/1.0/
xmlns=wav:http://ns.adobe.com/xmp/wav/1.0/
xmlns=xml:http://www.w3.org/XML/1998/namespace
xmlns=xmp:http://ns.adobe.com/xap/1.0/
xmlns=xmpBJ:http://ns.adobe.com/xap/1.0/bj/
xmlns=xmpDM:http://ns.adobe.com/xmp/1.0/DynamicMedia/
xmlns=xmpG:http://ns.adobe.com/xap/1.0/g/
xmlns=xmpGImg:http://ns.adobe.com/xap/1.0/g/img/
xmlns=xmpMM:http://ns.adobe.com/xap/1.0/mm/
xmlns=xmpNote:http://ns.adobe.com/xmp/note/
xmlns=xmpRights:http://ns.adobe.com/xap/1.0/rights/
xmlns=xmpT:http://ns.adobe.com/xap/1.0/t/
xmlns=xmpTPg:http://ns.adobe.com/xap/1.0/t/pg/
xmlns=xmpidq:http://ns.adobe.com/xmp/Identifier/qual/1.0/
959 rmills@rmillsmbp:~/gnu/exiv2/trunk $

```

#12 - 01 Sep 2015 20:40 - Robin Mills

rdf is pre-registered by the library:

```

959 rmills@rmillsmbp:~/gnu/exiv2/trunk $ exiv2 -vV -g rdf
exiv2 0.25 001900 (64 bit build)
xmlns=rdf:http://www.w3.org/1999/02/22-rdf-syntax-ns#
960 rmills@rmillsmbp:~/gnu/exiv2/trunk $

```

I think your test program does terminate gracefully when guarded by try .. catch. Sadly, you refused to follow my guidance on this subject.

This technology is built using contemporary C++ STL code. Throwing exceptions is part of that environment.

#13 - 01 Sep 2015 20:45 - Alan Pater

I found the namespaces for your file by running the command:

```
exiv2 -eX 321786_2.jpg
```

and then reading the resulting xmp sidecar file in a text editor. The XMP file contains the correct namespace URI's.

I did a quick search on the Interwebs and did not find the schemas for illustrator nor for fotoware.

#14 - 01 Sep 2015 21:43 - Robin Mills

- Status changed from Assigned to Closed

- % Done changed from 0 to 100

- Estimated time set to 4.00 h

#15 - 02 Sep 2015 09:02 - Harry McKame

Alan, Robin : Thanks for your answers and the wealth of information that they contain.

Note: None of your examples for exiv2.exe works for the release in exiv2-0.25-win.zip. Is it because I don't have Cygwin installed?

Robin:

I assure you that our production code is protected by try-catch. I wasn't aware that uncaught exceptions cause such weird crashes. I also confirm that with your dummy namespaces the test program works and the generated xmp is even correct.

So big question 1: Why do you crash when an unknown namespace is found?

Evidently, exiv2 can handle simple cases with its default code, so why not just invent a namespace string, say "http://dev.exiv2.org/illustrator", and just go on?

Until you do that, question 2: How can I detect which XMP tag is problematic before adding it to Exiv2::XmpData and before it crashes setXmpData() ?

FYI, our current use of Exiv2 is to replace software I have written for read/write of XMP, IPTC & EXIF. Yours is much more general and much better, although mine handles unknown namespaces without crashing - ;)

#16 - 02 Sep 2015 11:11 - Alan Pater

Don't use the dummy namespaces, use the official ones.

```
illustrator = "http://ns.adobe.com/illustrator/1.0/"
fwr = "http://ns.fotoware.com/iptcxmp-custom/1.0/"
fwr = "http://ns.fotoware.com/iptcxmp-reserved/1.0/"
```

#17 - 02 Sep 2015 11:21 - Harry McKame

Alan, sure I will, but this is just one image out of potentially hundreds of thousands coming from multiple sources.

I'm interested in a robust solution that will withstand anything I can throw at it, while still doing its job as much as possible, which is why I posted my above 2 questions.

#18 - 02 Sep 2015 12:16 - Robin Mills

Q1. The reason, I suggested adding try/catch was because of the message:

```
Unhandled exception at 0x761dc42d in test.exe: Microsoft C++ exception:
Exiv2::BasicError<char>; at memory location 0x01eef0c0.
```

Sure, Microsoft are angry. However they caught an exception from Exiv2.

I believe the exception is originally thrown by the Adobe XMP SDK

```
984 rmills@rmillsmbp:~/gnu/exiv2/trunk $ grep -H Throw xmpsdk/src/*.cpp | grep -i unregistered
xmpsdk/src/XMPCore_Impl.cpp:         XMP_Throw ( "Unregistered schema namespace URI", kXMPErr_BadSchema );
985 rmills@rmillsmbp:~/gnu/exiv2/trunk $
```

libexiv2 catches and throws an Exiv2 exception.

```
971 rmills@rmillsmbp:~/gnu/exiv2/trunk $ grep 'namespace info' src/*.cpp
src/error.cpp:         { 35, N_("No namespace info available for XMP prefix `%1'") }, // %1=prefix
972 rmills@rmillsmbp:~/gnu/exiv2/trunk $ grep -H throw src/*.cpp | grep 35
src/properties.cpp:         if (!xn) throw Error(35, prefix);
$
```

Because you haven't caught the exception, your program falls back on a Microsoft exception handler. By that time, your application stack or heap has been erased and all hell lets loose. It looks catastrophic. However they correctly identified that the exception was thrown by Exiv2.

Q2. You can catch the exception, register a namespace and try again. However the Adobe SDK does **NOT** do that. libexiv2 does **NOT** do that. However you can if you wish. *A word of caution however:* If you're going to invent URIs for unregistered namespaces and publish those files elsewhere, you could create chaos.

Adobe changed a prefix from xap to xmp and it has caused trouble (for us and others). It's very important to get the prefix/URI correct if anybody in the universe can access your file. [#751](#)

Throwing Exceptions is the C++/STL error mechanism (SEH = Structured Exception Handling.). I personally much prefer returning error codes. I've seldom found an engineer who agrees with my dislike of try/catch/throw.

I strongly encourage you to adopt libxiv2. The success of metadata workflows depends on standards compliant code to read and *more importantly write* metadata. Metadata **write** bugs will destroy workflows. Work with us because our code is mature, robust, cross-platform, tested, supported and actively developed. Alan and I and the rest of Team Exiv2 don't earn 1¢ from working on this project, so we're not motivated by self-interest. When we all work together, everybody gains. In software (1+1)=3

#19 - 02 Sep 2015 12:39 - Harry McKame

Robin,

For Q2: When my program sees a key and a value, and before it does :
gxmpData[ptag] = pvalue;
is there some way to find out if ptag belongs to an unknown schema?

If I can do that, I can recover from the error.

#20 - 02 Sep 2015 15:21 - Robin Mills

The simple way is to catch the exception:

```
try {
    gxmpData[ptag] = pvalue;
} catch (Exiv2::Error& e .... ) {
    if ( e something ) { // e tells you the name of the unregistered namespace
        XmpProperties::registerNs(a,b);
        gxmpData[ptag] = pvalue;
    }
}
```

Usually XMP tags are something like:

```
Xmp.illustrator.StartupProfile
```

In this case the *namespace prefix* is **illustrator**. However, namespaces can exist deeper in the key. For example:

```
Xmp.xmpMM.History[4]/stEvt:action
```

Uses the *namespace prefix* **xmpMM** and *namespace prefix* **stEvt** (both of which are preregistered):

```
1008 rmills@rmillsmbp:~/Downloads/Eiv2demo $ exiv2 -vV -g xmpMM -g stEvt
exiv2 0.25 001900 (64 bit build)
xmlns=stEvt:http://ns.adobe.com/xap/1.0/sType/ResourceEvent#
xmlns=xmpMM:http://ns.adobe.com/xap/1.0/mm/
1009 rmills@rmillsmbp:~/Downloads/Eiv2demo $
```

So the code has to allow for multiple exceptions to be thrown as several namespaces may be required. Something like this (**Caution** untested) code:

```
int tries = 0 ;
bool try_again = true ;
while ( try_again && tries++ < 100) {
    try_again = false ;
    try {
        gxmp[ptag] = pvalue ;
    } catch (Exiv2::error& e) {
        if ( e something ) { // e tells you the name of the unregistered namespace
            XmpProperties::registerNs(a,b);
            try_again=true;
        }
    }
}
```

#21 - 02 Sep 2015 15:26 - Robin Mills

- Status changed from Closed to Resolved

#22 - 02 Sep 2015 16:11 - Robin Mills

Whoops. I didn't answer your question about Cygwin/exiv2-0.25-win.zip I've never looked in that bundle. I think it is cross compiled for MinGW, so it should accept Dos filenames. However, I think you may need to \\ the path separators or something. For example: exiv2 -pa c:\\Eiv2Demo\\test.jpg

I think it's better to build your own binaries.

- If you're working in DOS/cmd.exe, build with msvc2005 (which builds with Visual Studio 2005/2008/10/12)
- If you're working in Cygwin (which I highly recommend), build with ./configure or CMake.
- If you're working in MinGW (which I don't recommend), build with ./configure.

My hesitation with MinGW is lack of experience with that system and not a criticism of MinGW. I only started supporting MinGW during the last 12 months.

#23 - 02 Sep 2015 16:39 - Harry McKame

Robin,

I think I have now all I need to continue advancing with Exiv2.

Thank you for your heroic support.

#24 - 03 Sep 2015 14:09 - Robin Mills

You're welcome.

Here's the bill: You go to open hub: <https://www.openhub.net/p/exiv2>, register (it's free) and click "I use this" for Exiv2 and give me some kudos. Or gimme some endorsements on LinkedIn.

#25 - 03 Sep 2015 17:57 - Harry McKame

I don't use much LinkedIn, but bill paid in full on OpenHUB, where I even posted the first review for Exiv2 (your name is there).

For the moment everything works fine, although the C++ framework is a bit restrictive. I have had to change our processing workflow to make it work with your design concepts.

The GPLv2 license is another problem. No idea why do you need or care for a restrictive license. We ourselves are months away from publishing this version of our software, so we will handle that when the time comes (currently using our own product).

#26 - 03 Sep 2015 18:53 - Robin Mills

Thanks. And very nice review. Nice to be thanked by name. There's a thing called Kudos on OpenHub - you can press the button on my Open Hub Page: <https://www.openhub.net/p/13985/contributors/60065117754297> Nobody's ever given me kudos. I don't expect it to be painful for either of us.

I'm surprised by your comment about C++. Perhaps we can discuss this further in a new thread on the forum. I was a Senior Computer Scientist at Adobe and all of Adobe's "core" libraries are C++ (search your copy of Adobe Reader for dlls/frameworks). I've never encountered a situation where a C++/DLL was an obstacle in product integration. All the common languages: perl, python, java, .Net, COM/DCOM have ways to call C++ libraries.

I don't know anything about GPLv2. Andreas, the project founder, set that up for Open Source use of libexiv2. If you have a commercial product, you can purchase a commercial license for Exiv2 and, as I understand it, your product will be totally free of any relationship to GPLv2. Andreas will provide details and cost (it's not expensive).

In providing support for Exiv2, it makes no difference to me if you are using an Open Source or Commercial License. I don't get paid in any way for the work I do for Exiv2.

#27 - 03 Sep 2015 20:01 - Robin Mills

Thanks, Harry. I got the Kudos on OpenHUB. Painless for me at least. Much appreciated.

#28 - 13 Oct 2015 19:53 - Robin Mills

- Status changed from Resolved to Closed

Files

Eiv2demo.7z	6.64 MB	31 Aug 2015	Harry McKame
Eiv2demo.7z	5.86 MB	01 Sep 2015	Harry McKame