Exiv2 - Bug #862

Video code is failing the test suite (on all plaforms)

19 Oct 2012 21:59 - Robin Mills

Status:	Closed	**Start date:**	19 Oct 2012
Priority:	Normal	**Due date:**	
Assignee:	Abhinav Badola	**% Done:**	100%
Category:	testing	**Estimated time:**	0.00 hour
Target version:	0.24		

Description

Two issues here:

1) The video code isn't passing its tests
2) I've assigned this issue to Category 'testing', however we need a Category 'video'

One of our users asked us to reduce the foot-print of our source download. So I've created a new tree in the depot to hold the video data. svn://dev.exiv2.org/svn/testdata/trunk. I've updated test/Makefile to download the video data on demand. This is discussed in #858http://dev.exiv2.org/issues/858

I've also update the test environment so we can run tests (from ./configure and CMake builds) using the command:
$ make tests

This doesn't include the video tests and these are run with the command:
$ make testv

I intend to do the same to the EPS tests which are now run with 'make teste'. I don't download the EPS files on demand yet, as this would break our published test procedure for 0.23.

Of course it's simple to re-instate testv/teste into 'make tests'. However because of the size of the download and because the testv is reporting errors, I'd prefer to keep your tests out of 'make tests'. The point is that because there's quite a lengthy report from the video tests, I often don't bother to look back in the transcript. So I could easily be missing important new errors from elsewhere. You'll find that running make testv expedites your tests as it omits all the other tests!

Abhinav:
I know you have a lot to do at college. Please let me know if I can help you in any way to get this investigated and resolved.

Associated revisions

Revision 2923 - 28 Oct 2012 14:29 - Robin Mills

Issue: #862. Please see discussions items 12 and 13 for more explanation.

Revision 2926 - 01 Nov 2012 19:50 - Robin Mills

Fix: #862 buffer overflow. See bug report discussion item#19.

History

#1 - 21 Oct 2012 06:01 - Abhinav Badola

Hi Robin,
I will look into the code asap and report back if I get stuck anywhere.

#2 - 23 Oct 2012 16:31 - Abhinav Badola

- File video-test.out added

Hi Robin,

I have detected the change that is creating all the mutations. It seems that the line-

const long bufMinSize = 4;

has been changed to

const long bufMinSize = 5;

in all occurrences in the video files.

For now, I have changed the code and pushed it back. The test is running fine on Linux.

Please update the test repository with the following attached video-test.out file, in the location - http://dev.exiv2.org/projects/exiv2/repository/show/testdata/trunk/video

It is only after that the test will run successfully.
Please report back to me, if you find any problem.

#3 - 23 Oct 2012 17:13 - Robin Mills

Thanks for investigating this, Abhinav. I'll build and test this after work this evening.

#4 - 24 Oct 2012 09:25 - Robin Mills

- File testunix.txt added

Abhinav

Thanks for doing this and things are much better. However, I'm still getting errors (both on Mac and Unix). I've attached the output files. Remarkably the output of the Mac (10.8.2/64 bit) and Unix (Kubuntu 12.10/64 bit) are different.

I see that you've changed some values of '5' to '4' in the code. I believe some of the '5's in the code arrived with Shawn's msvc patch which I submitted as SVN: 2879

For example QuickTimeVideo::decodeBlock() line#2920 QuickTimeVideo.cpp.

```
Before 2879 const long bufMinSize = 4; // your original code
SVN:   2879 const long bufMinSize = 5; // Shawn's change
SVN:   2920 const long bufMinSize = 4; // you've reverted
```

Shawn changed this because on line 655, you have

```
   buf.pData[4] = '\0'
```

Maybe the correct fix is to keep 2920 and change line 653. I haven't look at the constructor for DataBuf, it might alloc the nul terminator (however I doubt it).

```
DataBuf buf(bufMinSize);
to
DataBuf buf(bufMinSize+1);
```

The null terminator assignment is a litte odd because you're using memset on the next line. Perhaps we should change:

```
    void QuickTimeVideo::decodeBlock()
    {
        const long bufMinSize = 4;
        DataBuf buf(bufMinSize);
        unsigned long size = 0;
        buf.pData_[4] = '\0' ;

        std::memset(buf.pData_, 0x0, buf.size_);

        io_->read(buf.pData_, 4);
        if(io_->eof()) {
            continueTraversing_ = false;
            return;
        }

        size = Exiv2::getULong(buf.pData_, bigEndian);

        io_->read(buf.pData_, 4);
        if(size < 8)
            return;

        tagDecoder(buf,size-8);
    } // QuickTimeVideo::decodeBlock
```

to

```
    void QuickTimeVideo::decodeBlock()
    {
        const long bufMinSize = 4;
        DataBuf buf(bufMinSize+1);
        unsigned long size = 0;

        std::memset(buf.pData_, 0x0, buf.size_+1);

        io_->read(buf.pData_, bufMinSize);
        if(io_->eof()) {
            continueTraversing_ = false;
            return;
        }

        size = Exiv2::getULong(buf.pData_, bigEndian);

        io_->read(buf.pData_, bufMinSize);
        if(size < 8)
            return;

        tagDecoder(buf,size-8);
    } // QuickTimeVideo::decodeBlock
```

I'd like to ask you to:

1. Review the attached files from the Mac and Linux.
2. Consider my suggestion above.
3. Review every change in 2879.
4. Review the changes in 2920 to see if the restored '4's might cause buffer overflow.

I know this is rather tedious, however I believe this review will save time and bug reports in future.

#5 - 24 Oct 2012 16:30 - Robin Mills

- File testmac.txt added

#6 - 25 Oct 2012 15:28 - Abhinav Badola

- File video-test.out added

Hi Robin,

Please update the test repository with the following attached video-test.out file, in the location - http://dev.exiv2.org/projects/exiv2/repository/show/testdata/trunk/video
(I have attached a new revised file this time.)

It is only after that the test will run successfully. Without the above mentioned changes, the test will fail in Unix as well as Mac, as I have corrected some minor output discrepancies. I do not know how to update the above mentioned file, in the repository.

I have made small little changes in video files code as well. I hope the tests should now build successfully on both platforms.

#7 - 25 Oct 2012 16:26 - Robin Mills

Thanks, Abhinav. Very good and quick work. $ make testv -> "All testcases passed" on Kubuntu 12.10/64.

The test files now live at svn://dev.exiv2.org/svn/testdata/trunk and are downloaded on demand by test/Makefile when you run $ make testv in the <exiv2-dir>. I've submitted your revised file SVN:2922 (to testdata/trunk/video/video-test.out)

Have you reviewed the buffer size changes? I'm very nervous about line#655 in quicktimevideo.cpp

```
    void QuickTimeVideo::decodeBlock()
    {
        const long bufMinSize = 4;
        DataBuf buf(bufMinSize);
        unsigned long size = 0;
        buf.pData_[4] = '\0' ;    // really ?  Are you sure?

        std::memset(buf.pData_, 0x0, buf.size_);
```

Are you confident that we don't have a buffer overflow?

#8 - 26 Oct 2012 13:56 - Robin Mills

- File cygwin.out added

Abhinav. Apologies for spoiling your day. The tests are not passing on cygwin. Mac and Unix are OK. I haven't built/tested on the Windows/DevStudio builds. I attach cygwin.out.

#9 - 26 Oct 2012 14:05 - Abhinav Badola

Robin Mills wrote:

> Abhinav. Apologies for spoiling your day. The tests are not passing on cygwin. Mac and Unix are OK. I haven't built/tested on the Windows/DevStudio builds. I attach cygwin.out.

Hi Robin,
It seems that the code in the cygwin repository is old. I have removed the lines that would have created the following error report. Please check if the test was applied on the latest updated code.

#10 - 26 Oct 2012 15:00 - Robin Mills

Abhinav

You're not going to believe this. I totally reran the build and test. I pulled down a fresh copy of the trunk and built him with ./configure. Same result! There's something here. So I did a cmake build and it passes!

I'm going away for the weekend, however I'll be home on Sunday afternoon. If you don't have any ideas about this, I'll investigate. I'm sure it's nothing very much.

Thanks for looking at this so quickly.

Robin

#11 - 26 Oct 2012 15:15 - Abhinav Badola

Hi Robin,
Hope you have an awesome weekend. :)

Thanks for the info.
I am sure that I have removed the lines that were mentioned in the test output report.

There is one thing that I would like to point out here as well. For some reasons, I am not able to install exiv2 by normal method.

I mean in Linux, I tried the following commands -
[1] make config
[2] ./configure
[3] make -j4
[4] sudo make install

The compilation went on fine, but then installation showed an error -

```
libtool: install: /usr/bin/install -c -m 644 .libs/libexiv2.12.dylib /usr/local/lib/libexiv2.12.dylib
libtool: install: (cd /usr/local/lib && { ln -s -f libexiv2.12.dylib libexiv2.dylib || { rm -f libexiv2.dylib
&& ln -s libexiv2.12.dylib libexiv2.dylib; }; })
libtool: install: /usr/bin/install -c -m 644 .libs/libexiv2.lai /usr/local/lib/libexiv2.la
libtool: install: /usr/bin/install -c -m 644 .libs/libexiv2.a /usr/local/lib/libexiv2.a
libtool: install: chmod 644 /usr/local/lib/libexiv2.a
libtool: install: ranlib /usr/local/lib/libexiv2.a
----------------------------------------------------------------------
Libraries have been installed in:
   /usr/local/lib

If you ever happen to want to link against installed libraries
in a given directory, LIBDIR, you must either use libtool, and
specify the full pathname of the library, or use the `-LLIBDIR'
flag during linking and do at least one of the following:
   - add LIBDIR to the `DYLD_LIBRARY_PATH' environment variable
     during execution

See any operating system documentation about shared libraries for
more information, such as the ld(1) and ld.so(8) manual pages.
----------------------------------------------------------------------
../config/mkinstalldirs /usr/local/bin
../config/mkinstalldirs /usr/local/lib/pkgconfig
/usr/bin/install -c -m 644 ../config/exiv2.pc /usr/local/lib/pkgconfig/exiv2.pc
../config/mkinstalldirs /usr/local/bin
libtool: install: /usr/bin/install -c bin/exiv2 /usr/local/bin/exiv2
install: bin/exiv2: No such file or directory
make[1]: *** [install] Error 71
```

```
make: *** [install] Error 2
```

This error occurs when I execute the command

sudo make install

I think there is some modification in the Makefile or something, that may be the cause.

Note - The cmake method is working fine and installation is also working perfectly.

#12 - 26 Oct 2012 16:17 - Robin Mills

I haven't noticed what you've described. Do you have a directory /usr/local/bin ? Maybe something's disturbed ../config/mkinstalldirs which I think is create by the autotools. I'll also have a look at that on Sunday.

Well, I hope you also have a nice week-end. I'm off camping/running with my Adobe buddies. We go camping/running this weekend every year: http://clanmills.com/2011/BigSur/Saturday/

#13 - 28 Oct 2012 14:31 - Robin Mills

- Status changed from Assigned to Resolved

- % Done changed from 0 to 100

Well spotted, Abhinav. I've submitted a 3 byte fix that covers both! SVN:2923.

When I changed the build output directory to <exiv2dir>/bin and I missed building the exiv2 application into bin for the autotools build. So it didn't install, and the test suite was running the "old" version of exiv2. This also explains why it passes on the cmake build, which did output correctly to <exiv2dir>/bin.

I've changing the status to "Resolved" and 100% Done as I believe this is now complete. Thanks very much for working on this. We won't close the issue until the end of 0.24 development - just in case something else appears concerning this.

#14 - 28 Oct 2012 14:43 - Abhinav Badola

Cool. :)

Yeah now the bug seems fixed. I have downloaded and tested the new code and it installed successfully.
Thanks. :)

#15 - 29 Oct 2012 19:50 - Robin Mills

Abhinav

Everything looks good. May I ask you again about possible buffer overflows:

```
    void QuickTimeVideo::decodeBlock()
    {
        const long bufMinSize = 4;
        DataBuf buf(bufMinSize);
        unsigned long size = 0;
        buf.pData_[4] = '\0' ;    // really ?  Are you sure?

        std::memset(buf.pData_, 0x0, buf.size_);
```

Are you OK with this?

#16 - 30 Oct 2012 15:17 - Abhinav Badola

Hi Robin,

```
const long bufMinSize = 4;
DataBuf buf(bufMinSize);
buf.pData_[4] = '\0' ;
```

This has been used just as a precautionary measure.

Actually, the buffer size can't be changed and we cant include "\0" into the data buffer as well, because that would lead to conversion discrepancies, especially in case of Integer or Float.
I can't say it for sure that it will never create any buffer overflow, but it has worked fine with all the test cases that I took under consideration. I have also applied checks at places in the code to prevent it from breaking down.
So I think it would be safe to consider that it won't cause any buffer overflow kind of problem.

#17 - 30 Oct 2012 16:02 - Robin Mills

Would it hurt to change:

```
DataBuf buf(bufMinSize);
```

to:

```
DataBuf buf(bufMinSize+1);
```

And review similar changes in Shawn's submission?

#18 - 31 Oct 2012 02:13 - Abhinav Badola

Robin Mills wrote:

> Would it hurt to change:
> [...]
> to:
> [...]
> And review similar changes in Shawn's submission?

Yes, it will lead to a lot of misinterpretations(when the buffer is interpreted to integer or float) and may even break the code.
The data generated by interpreting the buffer would not be accurate and would also cause all of the video tests to fail.

One of the prominent reason that I could think about is -
[1] The binary data is stored either in big endian or small endian format. So the interpretation differs from case to case. So if the buffer size is increased to five (instead of four), the interpretation may be correct for little endian, but may fail in the case if big endian or vice versa. The conversion function would be expecting four byte, where it would get five bytes instead and then the result may not be well defined.

[2] The '\0' is important and can't be removed from the code either. This is because sometimes the buffer is interpreted as character string. And it can't be placed inside the buffer, cause then when a four character string is stored in the buffer, it will get overwritten, and will be useless.

#19 - 01 Nov 2012 19:49 - Robin Mills

- File debug_error.png added

When I started testing, the MSVC debug builds (32/64 bit DLL/Static) all asserted and displayed a dialog box about heap corruption for a couple of the test files (*.avi for certain).

debug_error.png
I've fixed this "in-line" by adding +1 to the DataBuf constructors. I can't help but feel there's a better solution in modifying the constructor itself. I considered two ideas:

1) Adding options inputs to the constructor:
DataBuf buff(long size,bool bAddNulTerminator = false,bool bZeroFillBuffer = false)

2) If you request allocate 8 or less bytes you get 8 zero'd bytes.

However I've stuck with the in-line fix as it passed the test suite and it's "good enough". It also seems better to have this exposed in the code and not hidden away. However there's a risk that I haven't spotted every instance. I know Abhinav will review the changes.

#20 - 24 Jul 2013 15:19 - Robin Mills

- Status changed from Resolved to Closed

Fixed in 0.24.

Files

video-test.out	17.6 KB	23 Oct 2012	Abhinav Badola
testunix.txt	3.64 KB	24 Oct 2012	Robin Mills
testmac.txt	3.82 KB	24 Oct 2012	Robin Mills
video-test.out	17.6 KB	25 Oct 2012	Abhinav Badola
cygwin.out	8.96 KB	26 Oct 2012	Robin Mills
debug_error.png	41.9 KB	01 Nov 2012	Robin Mills