

## Exiv2 - Bug #1007

### exiv2 0.24 Build error on blackfin arch

03 Dec 2014 14:02 - Nicolas Serafini

<b>Status:</b>	Closed	<b>Start date:</b>	03 Dec 2014
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Robin Mills	<b>% Done:</b>	100%
<b>Category:</b>	build	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	0.25		

**Description**

exiv2 0.24 is included as a package into buildroot and we have a build error on a blackfin arch with the blackfin uclinux 2014R1 toolchain.

The build output is here <http://autobuild.buildroot.net/results/09d/09d9b525eeee458247e10a7948070c00bbabb964/build-end.log>

It seems that the inclusion of link.h in version.cpp line 119 create this error. Perhaps there is a problem into the toolchain because the error comes from some other included files in "link.h" but if I remove the inclusion of "link.h" all seems to work fine even with another toolchain (linaro 2014.07).

Do we really need to include this file ?

#### Associated revisions

##### Revision 3397 - 03 Dec 2014 19:49 - Robin Mills

#1007. Thank You, Nicolas for reporting this and identifying the fix.

#### History

##### #1 - 03 Dec 2014 14:06 - Nicolas Serafini

oops it's linaro 2014.09 and not 2014.07

##### #2 - 03 Dec 2014 15:08 - Robin Mills

- *Tracker changed from Bug to Support*
- *Status changed from New to Assigned*
- *Assignee set to Robin Mills*
- *Target version set to 0.25*

Ah, there's something in your distribution of linux that causing version.cpp to fail to compile.

I think your distribution either doesn't include link.h, or is remarkably different from that included in Ubuntu and Fedora, which are my linux test/build distros.

```
#elif defined(__linux__)
# include <unistd.h>
// http://syprog.blogspot.com/2011/12/listing-loaded-shared-objects-in-linux.html
# include "link.h"
# include <dlfcn.h>
struct something
{
    void* pointers[3];
    struct something* ptr;
};
struct lmap
{
    void* base_address; /* Base address of the shared object */
    char* path; /* Absolute file name (path) of the shared object */
    void* not_needed1; /* Pointer to the dynamic section of the shared object */
    struct lmap *next, *prev; /* chain of loaded objects */
};
#endif
...
```

```

#elif defined(__linux__)
    // http://stackoverflow.com/questions/606041/how-do-i-get-the-path-of-a-process-in-unix-linux
    char proc[100];
    char path[500];
    sprintf(proc, "/proc/%d/exe", getpid());
    int l = readlink (proc, path, sizeof(path)-1);
    if (l>0) {
        path[l]=0;
        libs.push_back(path);
    } else {
        libs.push_back("unknown");
    }
}

// http://syprog.blogspot.com/2011/12/listing-loaded-shared-objects-in-linux.html
struct lmap*      pl;
void*             ph = dlopen(NULL, RTLD_NOW);
struct something* p = (struct something*) ph;

p = p->ptr;
pl = (struct lmap*)p->ptr;

while ( pl )
{
    libs.push_back(pl->path);
    pl = pl->next;
}
#endif

```

The purpose of this code is to enumerate libraries being used by exiv2. This is inspected by our test suite to be validate we are using the correct libraries. Here's the output from gubuntu:

```

1001 rmills@rmillssmbp-kubuntu:~ $ which exiv2
/usr/local/bin/exiv2
1002 rmills@rmillssmbp-kubuntu:~ $ exiv2 -v -V
exiv2 0.24 001800 (64 bit build)
Copyright (C) 2004-2013 Andreas Huggel.

```

```

This program is free software; you can redistribute it and/or
...
Boston, MA 02110-1301 USA
exiv2=0.24.0
platform=linux
compiler=G++
bits=64
dll=1
debug=0
version=4.8.2
date=Oct 7 2014
time=21:39:32
svn=3372
executable=/usr/local/bin/exiv2
library=/usr/local/lib/libexiv2.so.13
library=/usr/lib/x86_64-linux-gnu/libstdc++.so.6
library=/lib/x86_64-linux-gnu/libgcc_s.so.1
library=/lib/x86_64-linux-gnu/libc.so.6
library=/usr/local/lib/libz.so.1
library=/usr/local/lib/libexpat.so.1
library=/lib/x86_64-linux-gnu/libdl.so.2
library=/lib/x86_64-linux-gnu/libm.so.6
library=/lib64/ld-linux-x86-64.so.2
1003 rmills@rmillssmbp-kubuntu:~ $

```

I'm not sure what blackfin arch with the blackfin 2014.09 is. Presumably, I can google, download and create a virtual machine to run this. Any pointers?

The code has been constructed to deal with GCC/Clang/Windows/Cygwin/Mac/Linux/GNU platforms and I'm quite certain that it can be modified to deal with your distribution. However, it will involve a change to the CMake\*.txt files or version.cpp. I don't think we're going to be able to fix this from the build command line. Does this create difficulties for you?

### #3 - 03 Dec 2014 16:27 - Nicolas Serafini

Thanks for your response.

Blackfin is a processor from analog devices on which we build a full linux distribution. It's possible that the toolchain used has a bug because the build

crash into a file from the libC (uclibc) used by the toolchain.

But I have just tested to compile on my desktop after removing the line `#include "link.h"` at line 119 and it work's fine without.

```
nse@nse-ubuntu ~/temp/exiv2-0.24/bin » ./exiv2 -v -V
```

```
127 ↵
exiv2 0.24 001800 (64 bit build)
Copyright (C) 2004-2013 Andreas Huggel.

This program is free software; you can redistribute it and/or
...
Boston, MA 02110-1301 USA
exiv2=0.24.0
platform=linux
compiler=G++
bits=64
dll=1
debug=0
version=4.8.2
date=Dec 3 2014
time=16:20:29
executable=/home/nse/temp/exiv2-0.24/bin/exiv2
library=/home/nse/temp/exiv2-0.24/src/libexiv2.so.13
library=/usr/lib/x86_64-linux-gnu/libstdc++.so.6
library=/lib/x86_64-linux-gnu/libgcc_s.so.1
library=/lib/x86_64-linux-gnu/libc.so.6
library=/lib/x86_64-linux-gnu/libdl.so.2
library=/lib/x86_64-linux-gnu/libexpat.so.1
library=/lib/x86_64-linux-gnu/libm.s
```

#### #4 - 03 Dec 2014 17:24 - Robin Mills

Nicolas:

Can you suggest how I can hide link.h from your build. Something like:

```
#elif defined(__linux__)
# include <unistd.h>
// http://syprog.blogspot.com/2011/12/listing-loaded-shared-objects-in-linux.html
# ifndef __Blackfin__
# include "link.h"
# endif
# include <dlfcn.h>
```

I will submit your recommendation to the trunk and you'll be future proofed.

If you're OK with this, I'd like to mark this resolved. Let me know.

Robin

#### #5 - 03 Dec 2014 18:37 - Nicolas Serafini

No I don't want an architecture flag to exclude this header. If the toolchain has a bug when this header is included it's from the toolchain maintainer to fix this.

For the exiv2 part I think the inclusion of link.h is not needed on any linux distribution.

I have build two times using cmake and configure a version for my Desktop on ubuntu without the include of link.h and it works.

In version.cpp no symbol from the link.h header is used.

version.cpp create only a minimal link\_map structure and the web page from where the example come doesn't specify the need to include this header.

For me the patch is

```
#elif defined(__linux__)
# include <unistd.h>
// http://syprog.blogspot.com/2011/12/listing-loaded-shared-objects-in-linux.html
# include <dlfcn.h>
```

#### #6 - 03 Dec 2014 19:52 - Robin Mills

- *Tracker changed from Support to Bug*

- *Status changed from Assigned to Resolved*

Fix submitted [r3397](#). Thank You, Nicolas, both for reporting this and for the fix. Much Appreciated.

**#7 - 08 May 2015 16:32 - Robin Mills**

- % Done changed from 0 to 100

**#8 - 21 Jun 2015 16:39 - Andreas Huggel**

- Status changed from Resolved to Closed