

Public Support - Support Request #4249

Failed to compile components of ADTF File Library on Ubuntu

2018-09-24 18:00 - hidden

Status: Closed	
Priority: Normal	
Category:	
Customer: DAIMLER	Product Issue Numbers:
Department:	Affected Products: ADTF File Library 0.2.1 (BETA)
Requester's Priority: Normal	Platform: Ubuntu 16.04 64bit, Ubuntu 18.04 64bit
Support Level: 3rd Level	Topic: FileLibrary::Common
Resolution: Solved Issue	FAQ Links:
Description	
Supportanfrage	
<p>ich will die adtf-file-library 0.2.1 auf einem Ubuntu system so wie im README angegeben kompilieren. Leider schlägt das Kompilieren der ddl Library fehl.</p>	
<p>Hier die ausgeführten Schritte:</p>	
<ul style="list-style-type: none">• System Ubuntu 16.04.5 LTS• adtf_file-0.2.1.zip, a_util-5.1.0.zip, ddl-4.0.3.zip aus dem Download Center von digitalwerk.net heruntergeladen• cmake 3.12 von cmake.org binary runtergeladen und separat installiert (a_util erfordert mind. 3.8)• a_util-5.1.0 mit cmake und make kompiliert und installiert nach /usr/local/adtf-toolboxes/filelib/0.2.1• ddl-4.0.3: Makefile erstellt mit dem gleichen CMAKE_INSTALL_PREFIX wie a_util	
<p>Beim Kompilieren mit make bekommt man folgenden Fehler:</p>	
<pre>[1]heinrich@ubuntu:~/adtf_file_lib-0.2.1/ddl-4.0.3/Debug\$ make Scanning dependencies of target ddl [1%] Building CXX object CMakeFiles/ddl.dir/ddlrepresentation/ddl_error.cpp.o [2%] Building CXX object CMakeFiles/ddl.dir/ddlrepresentation/ddlversionhelper.cpp.o [4%] Building CXX object CMakeFiles/ddl.dir/ddlrepresentation/ddl.cpp.o [5%] Building CXX object CMakeFiles/ddl.dir/ddlrepresentation/ddlalignment.cpp.o [6%] Building CXX object CMakeFiles/ddl.dir/ddlrepresentation/ddlbyteorder.cpp.o [8%] Building CXX object CMakeFiles/ddl.dir/ddlrepresentation/ddlcomplex.cpp.o [9%] Building CXX object CMakeFiles/ddl.dir/ddlrepresentation/ddldatatype.cpp.o [10%] Building CXX object CMakeFiles/ddl.dir/ddlrepresentation/ddlheader.cpp.o [12%] Building CXX object CMakeFiles/ddl.dir/ddlrepresentation/ddlstream.cpp.o [13%] Building CXX object CMakeFiles/ddl.dir/ddlrepresentation/ddlenum.cpp.o [15%] Building CXX object CMakeFiles/ddl.dir/ddlrepresentation/ddlcontainer.cpp.o [16%] Building CXX object CMakeFiles/ddl.dir/ddlrepresentation/ddldescription.cpp.o [17%] Building CXX object CMakeFiles/ddl.dir/ddlrepresentation/ddlunit.cpp.o [19%] Building CXX object CMakeFiles/ddl.dir/ddlrepresentation/ddlrefunit.cpp.o</pre>	

```
[ 20%] Building CXX object
CMakeFiles/ddl.dir/ddlrepresentation/ddlprefix.cpp.o
[ 21%] Building CXX object
CMakeFiles/ddl.dir/ddlrepresentation/ddlxtdeclaration.cpp.o
[ 23%] Building CXX object
CMakeFiles/ddl.dir/ddlrepresentation/ddlelement.cpp.o
[ 24%] Building CXX object
CMakeFiles/ddl.dir/ddlrepresentation/ddlbaseunit.cpp.o
[ 26%] Building CXX object
CMakeFiles/ddl.dir/ddlrepresentation/ddlstreamstruct.cpp.o
[ 27%] Building CXX object
CMakeFiles/ddl.dir/ddlrepresentation/ddlprinter.cpp.o
[ 28%] Building CXX object
CMakeFiles/ddl.dir/ddlrepresentation/ddlimporter.cpp.o
[ 30%] Building CXX object
CMakeFiles/ddl.dir/ddlrepresentation/ddlcloner.cpp.o
[ 31%] Building CXX object
CMakeFiles/ddl.dir/ddlrepresentation/ddlresolver.cpp.o
[ 32%] Building CXX object
CMakeFiles/ddl.dir/ddlrepresentation/ddlrepair.cpp.o
[ 34%] Building CXX object
CMakeFiles/ddl.dir/ddlrepresentation/ddlinspector.cpp.o
[ 35%] Building CXX object
CMakeFiles/ddl.dir/ddlrepresentation/ddlcompare.cpp.o
/home/heinrich/adtf_file_lib-0.2.1/ddl-4.0.3/ddlrepresentation/ddlcompare.cpp:
In static member function 'static a_util::result::Result
ddl::cDDLCompare::IsBinaryEqual(const string&, const string&, const string&,
const string&, bool)':
/home/heinrich/adtf_file_lib-0.2.1/ddl-4.0.3/ddlrepresentation/ddlcompare.cpp:105:5:
error: 'unique_ptr' is not a member of 'std'
    std::unique_ptr<cDDLDescription>
pRefDesc(cDDLDescription::CreateDefault())
```

```
/home/heinrich/adtf_file_lib-0.2.1/ddl-4.0.3/ddlrepresentation/ddlcompare.cpp:105:5:
```

```
note: suggested alternatives:
```

```
In file included from
```

```
/usr/local/adtf-toolboxes/filelib/0.2.1/include/a_util/memory/unique_ptr.h:33:0,
```

```
from
```

```

    from
/usr/local/adtf-toolboxes/filelib/0.2.1/include/a_util/filesystem/path.h:20,
    from
/usr/local/adtf-toolboxes/filelib/0.2.1/include/a_util/filesystem.h:20,
    from /home/heinrich/adtf_file_lib-0.2.1/ddl-4.0.3/ddl.h:46,
    from
/home/heinrich/adtf_file_lib-0.2.1/ddl-4.0.3/ddlrepresentation/ddlcompare.cpp:17:
/usr/local/adtf-toolboxes/filelib/0.2.1/include/a_util/memory/detail/unique_ptr_fallback_decl.h:35:7:
note: 'a_util::memory::detail::unique_ptr'
class unique_ptr
    ^
/usr/local/adtf-toolboxes/filelib/0.2.1/include/a_util/memory/detail/unique_ptr_fallback_decl.h:35:7:
note: 'a_util::memory::detail::unique_ptr'
/home/heinrich/adtf_file_lib-0.2.1/ddl-4.0.3/ddlrepresentation/ddlcompare.cpp:105:36:
error: expected primary-expression before '>' token
std::unique_ptr<cDDLDescription>
pRefDesc(cDDLDescription::CreateDefault())
    ^
/home/heinrich/adtf_file_lib-0.2.1/ddl-4.0.3/ddlrepresentation/ddlcompare.cpp:105:79:
error: 'pRefDesc' was not declared in this scope
std::unique_ptr<cDDLDescription>
pRefDesc(cDDLDescription::CreateDefault());

```

```

CMakeFiles/ddl.dir/build.make:662: recipe for target
'CMakeFiles/ddl.dir/ddlrepresentation/ddlcompare.cpp.o' failed
make2: * [CMakeFiles/ddl.dir/ddlrepresentation/ddlcompare.cpp.o] Error 1
CMakeFiles/Makefile2:67: recipe for target 'CMakeFiles/ddl.dir/all' failed
make1: [CMakeFiles/ddl.dir/all] Error 2
Makefile:127: recipe for target 'all' failed
make: ** [all] Error 2

```

Können Sie hier unterstützen?

Lösung

Das Problem ist ein fehlerhaftes Verhalten der CMakeFiles der a_utils mit CMake 3.12. Man muss bei der 5.1 leider zwingend Cmake 3.9.x einsetzen, sonst klappt es nicht. Laut AEV ist das bei neueren a_utils gefixed. Also ich habs letztendlich mit Cmake 3.9.6 und folgendem Aufruf hinbekommen, bitte bei allen drei Projekten (a_utils, ddl, adtf_file) verwenden:

```

~/devel/cmake-3.9.6-Linux-x86_64/bin/cmake -DCMAKE_CXX_STANDARD=14 -DCMAKE_INSTALL_PREFIX=<blah> -
DCMAKE_BUILD_TYPE=Debug -Da_util_cmake_enable_documentation=0 <src>

```

History

#1 - 2018-09-25 08:17 - hidden

- Project changed from Public Support to 9
- Status changed from New to In Progress
- Topic set to FileLibrary::Common
- Customer set to DAIMLER
- Affected Products ADTF File Library 0.2.1 (BETA) added
- Platform Ubuntu 16.04 64bit added

#2 - 2018-09-25 10:09 - hidden

- Description updated

@Martin: Kannst du mal bitte drauf schauen ? Ich kann das nicht nachstellen

#3 - 2018-09-27 16:22 - hidden

Ich hab mir jetzt eine jungfräuliche VM aufgesetzt, und dort kann ich das Problem nachstellen. Ich habs auch mit CMAKE_CXX_STANDARD=11 probiert, aber da kann ich schon die a_util nicht kompilieren. beim gcc 5.4 ist noch c++98 der default. Wir müssten die Flags rausfinden die wir beim Erstellen der Libs (aller drei) mit Conan verwenden, da steckt ich aber nicht drin. @Flo, kannst Du das an einen conan-erfahrenen weitergeben?

#4 - 2018-10-01 08:27 - hidden

- Support Level changed from 2nd Level to 3rd Level

#5 - 2018-10-02 10:20 - hidden

Das Problem ist ein fehlerhaftes Verhalten der CMakeFiles der a_utils mit CMake 3.12. Man muss bei der 5.1 leider zwingend Cmake 3.9.x einsetzen, sonst klappt es nicht. Laut AEV ist das bei neueren a_utils gefixed. Also ich hab's letztendlich mit Cmake 3.9.6 und folgendem Aufruf hinbekommen, bitte bei allen drei Projekten (a_utils, ddl, adtf_file) verwenden:

```
~/devel/cmake-3.9.6-Linux-x86_64/bin/cmake -DCMAKE_CXX_STANDARD=14 -DCMAKE_INSTALL_PREFIX=<blah> -DCMAKE_BUILD_TYPE=Debug -Da_util_cmake_enable_documentation=0 <src>
```

#6 - 2018-10-02 11:08 - hidden

- Status changed from In Progress to Customer Feedback Required

#7 - 2018-10-08 13:21 - hidden

Hallo Michael,

hat dir das weitergeholfen ?

#8 - 2018-10-10 08:00 - hidden

Hallo Florian,

ja das hat weitergeholfen. Ich konnte erfolgreich kompilieren unter Ubuntu16 und Ubuntu14.

ABER es besteht die Anforderung, dies unter Ubuntu18 zu kompilieren. Dabei ist beim Kompilieren folgendes aufgefallen:

Es gibt beim Kompilieren der ddl folgenden Fehler:

```
[ 83%] Building CXX object
ddlgenerators/generator_library/CMakeFiles/ddl_generator.dir/headerrepresentation/header_importer.cpp.o
/home/miheintr/proj/adtf_file_lib-0.2.1/ddl-4.0.3/ddlgenerators/generator_library/headerrepresentation/header_importer.cpp:
In member function 'a_util::result::Result
ddl::cHeaderImporter::AddErrorDescription(const string&, const char*)':
/home/miheintr/proj/adtf_file_lib-0.2.1/ddl-4.0.3/ddlgenerators/generator_library/headerrepresentation/header_importer.cpp:1166:30:
error: ISO C++ forbids comparison between pointer and integer [-fpermissive]
    while (pSearchPos != '\0')
           ^~~~
ddlgenerators/generator_library/CMakeFiles/ddl_generator.dir/build.make:254:
recipe for target
'ddlgenerators/generator_library/CMakeFiles/ddl_generator.dir/headerrepresentation/header_importer.cpp.o'
failed
make2:
[ddlgenerators/generator_library/CMakeFiles/ddl_generator.dir/headerrepresentation/header_importer.cpp.o]
Error 1
CMakeFiles/Makefile2:146: recipe for target
'ddlgenerators/generator_library/CMakeFiles/ddl_generator.dir/all' failed
make1: *
[ddlgenerators/generator_library/CMakeFiles/ddl_generator.dir/all] Error 2
Makefile:129: recipe for target 'all' failed
make: ** [all] Error 2
```

Der Daimler-Kollege meint, dass dies auch ein Bug sein könnte. Ich zitiere seinen Kommentar:

ich habe eben erfolgreich unter Ubuntu 18.04 gebaut, nachdem ich die Zeile

```
while (pSearchPos != '\0')
```

durch

```
while (*pSearchPos != '\0')
```

ersetzt habe.

Ich denke, dass diese Änderung den Code nicht nur mit Compiler von 18.04 kompatibel macht, sondern einen echten Bug behebt. Kann der Support von Digitalwerk dies bestätigen?

Der letzten Frage schließe ich mich an und bedanke mich im voraus.

Mit freundlichen Grüßen,
Michael Heinrich

#9 - 2018-10-10 08:21 - hidden

- Status changed from *Customer Feedback Required* to *In Progress*
- Platform *Ubuntu 18.04 64bit* added

Hallo Michael,

danke für die Rückmeldung, ich denke auch das dies ein Bug ist, der leider aber erst vom neuen Compiler "gefunden" wird. Allerdings muss ich auch dazu sagen, dass wir hier ausschließlich auf Ubuntu 16.04 entwickeln und auch supporten, unabhängig davon, dass es im Falle eines Bugs natürlich gefixt werden muss. Nur als Hinweis, dass wir weitere Quereffekte auf 18.04 nicht ausschließen können und keinesfalls supporten.

@Martin: Siehst du das genauso ? Wenn ja, sollten wir ein Ticket machen.

#10 - 2018-10-10 08:23 - hidden

PS: Die ADTF File Library ist Open Source zu verstehen und wir auch als Git Clone geliefert. Bei derartigen Funden und/oder Verbesserungsvorschlägen kann jederzeit ein Diff geschickt werden, eine Mitarbeit ist nicht gefordert aber willkommen. Ob es integriert wird muss natürlich entschieden werden.

#11 - 2018-10-15 08:52 - hidden

Hi!

Danke für die Rückmeldung, das war natürlich ein Bug. Flo hast Du da ein Ticket erstellt? Ich denke aber, dass der in neueren ddl Versionen schon gefixt ist, da die AEV da auch mit neueren Compilern getestet?

Die adtf_file Funktionalität betrifft der Fehler zum Glück nicht, da der header import dort nirgends Verwendung findet.

Also ich persönlich bin auf jeden Fall dafür, dass wir auch Compiler spezifische Probleme (also bei neueren gcc und msvc Versionen) beheben, da dass der Verbreitung sicher immer zutunlich ist und die integration der Library in neue Projekte fördert.
Grüße,

Martin

#12 - 2018-10-15 12:44 - hidden

- Project changed from *9* to *Public Support*
- Subject changed from *[Ticket#2018091210000945] Install Software: ADTF 3 File Library* to *Failed to compile components of ADTF File Library on Ubuntu*
- Description updated
- Status changed from *In Progress* to *To Be Closed*
- Private changed from *Yes* to *No*
- Resolution set to *Solved Issue*

Hallo zusammen,

damit kann das Ticket geschlossen werden. Im aktuellen Master der DDL ist das auch schon entsprechend angepasst wie Martin meinte, damit ist es in der kommenden ADTF File Lieferung auch gefixt.

Wir können das Support Ticket an dieser Stelle schließen.

#13 - 2018-10-15 12:45 - hidden

- Status changed from *To Be Closed* to *Closed*