

Public Support - Support Request #6357

ADTF adtf_file example, no type deserializer

2019-03-01 14:00 - hidden

Status:	Closed		
Priority:	Normal		
Category:			
Customer:	AUDI	Product Issue Numbers:	https://www.cip.audi.de/jira/projects/CDI/FHD/issues/CDIFHD-24
Department:	EFS	Affected Products:	ADTF 3.3.1
Requester's Priority:	Normal	Platform:	Windows 7 64bit
Support Level:	3rd Level	Topic:	ADTF::Common
Resolution:	Solved Issue	FAQ Links:	

Description

Support Anfrage:

ich versuche gerade mit ADTF 3.3.1 und dem "adtf_file" example\fileaccess ein .dat file zu öffnen.
Compilierung des examples über cmake und Visual Studio hat soweit auch funktioniert, .exe wurde erstellt.
Beim Aufruf der .exe mit einem ADTF .dat file als Argument bekomme ich aber folgende Fehlermeldung:

```
F:\adtf\3.3.1\3rdparty\adtf_file\build\src\examples\fileaccess\Debug>fileaccess.exe F:\adtf_example_testdata\test.dat test.csv
could not query info of file F:\adtf_example_testdata\test.dat
no type deserializer for 'adtf.type.flexray.adtf2_support.serialization.adtf.cid' available
```

Es fehlt wohl der flexray deserializer. Leider komme ich ab diesem Punkt selbst nicht mehr weiter.
Können sie mich bitte unterstützen?

Lösung:

Es wird dringend empfohlen auf die aktuellen Versionen von ADTF (ADTF 3.5.0) bzw. in diesen Fall der ADTF File Library 0.4.0 (BETA) zu wechseln.
Das ADTF [File Access Example](#) ist in der aktuellen Version in der SDK Doku der ADTF File Library hinterlegt. Siehe Link oben oder [hier](#) .
Seit ADTF 3.4.0 ist das DAT Handling endgültig in die ADTF File Library übergegangen.
Siehe hierzu auch die entsprechende [Release Information ADTF 3.4.0](#)

Move ADTF DAT Processing and tooling to ADTF File Library (delivered by 3rd Party ifhd)

Inzwischen ist auch die Device Toolbox 3.0.0 verfügbar (Stichwort Flexray), siehe [Release Information Device Toolbox 3.0.0](#)

Um das eigentliche Problem zu lösen, muss das Example so erweitert werden, dass weitere adtffileplugins nachladen werden können (darüber kannst man Deserializer bekannt machen). Dazu gibt es auch ein Ticket (CDIFHD-24). Mit der Device kannst man die typen aus ADTF 2.x bekannt machen, man müsste das Example in etwa so erweitern:

```
// ...

#include <map>

// initialize ADTF File and Plugin Mechanism
static adtf_file::Objects oObjects;
static adtf_file::PluginInitializer oInitializer([]
{
    adtf_file::add_standard_objects();
});

void query_file_info(const std::string& filename)
{
    using namespace adtf_file;

    // load deserializer from adtffileplugin
```

```

    adtf_file::load_plugin(
"<path_to_device_toolbox_3_0_0>/bin/adtf_devtb_2_deserializer.adtffileplugin");
// or debug: adtf_file::load_plugin("<path_to_device_toolbox_3_0_0>/bin/debug/adtf_devtb_2_deseria
lizerd.adtffileplugin");

    // open file -> create reader from former added settings
    adtf_file::Reader reader(filename,
        adtf_file::getFactories<adtf_file::StreamTypeDeserializers,
            adtf_file::StreamTypeDeserializer>(),
        adtf_file::getFactories<adtf_file::SampleDeserializerFactories,
            adtf_file::SampleDeserializerFactory>(),
        std::make_shared<adtf_file::sample_factory<adtf_file::DefaultSample>>(),
        std::make_shared<adtf_file::stream_type_factory<adtf_file::DefaultStreamType>>
    ));

    std::cout << "File: " << filename << std::endl;

    // ...

void access_file_data(const std::string& filename, const std::string& csv_file_path)
{
    using namespace adtf_file;

    // load deserializer from adtffileplugin
    adtf_file::load_plugin(
"<path_to_device_toolbox_3_0_0>/bin/adtf_devtb_2_deserializer.adtffileplugin");
// or debug: adtf_file::load_plugin("<path_to_device_toolbox_3_0_0>/bin/debug/adtf_devtb_2_deseria
lizerd.adtffileplugin");

    adtf_file::Reader reader(filename,
        adtf_file::getFactories<adtf_file::StreamTypeDeserializers,
            adtf_file::StreamTypeDeserializer>(),
        adtf_file::getFactories<adtf_file::SampleDeserializerFactories,
            adtf_file::SampleDeserializerFactory>(),
        std::make_shared<adtf_file::sample_factory<adtf_file::DefaultSample>>(),
        std::make_shared<adtf_file::stream_type_factory<adtf_file::DefaultStreamType>>
    ));

    StreamsInfo stream_info(reader);

```

Patch an dieser Stelle nicht vergessen (siehe Device Toolbox Download), sofern die ADTF File Library 0.4.0 (= Basis ab ADTF 3.4.0) verwendet wird (dringend empfohlen)

History

#1 - 2019-03-01 14:06 - hidden

- File deleted (image003.png)

#2 - 2019-03-01 14:07 - hidden

- File deleted (image005.png)

#3 - 2019-03-01 14:07 - hidden

- File deleted (image002.png)

#4 - 2019-03-04 08:53 - hidden

- Project changed from Public Support to 11
- Status changed from New to In Progress
- Topic set to ADTF::Common
- Customer set to AUDI
- Department set to EFS
- Affected Products ADTF 3.3.1 added
- Platform Windows 7 64bit added

#5 - 2019-03-04 09:29 - hidden

- Status changed from In Progress to Customer Feedback Required

Hallo Christian Belsito,

zuerst würde wir dringend empfehlen auf die aktuellen Versionen von ADTF (ADTF 3.5.0) bzw. in diesen Fall der ADTF File Library 0.4.0 (BETA) zu wechseln.

Das ADTF [File Access Example](#) ist in der aktuellen Version in der SDK Doku der ADTF File Library hinterlegt. Siehe Link oben oder [hier](#) .

Seit ADTF 3.4.0 ist das DAT Handling endgültig in die ADTF File Library übergegangen.

Siehe hierzu auch die entsprechende [Release Information ADTF 3.4.0](#)

Move ADTF DAT Processing and tooling to ADTF File Library (delivered by 3rd Party ifhd)

Inzwischen ist auch die Device Toolbox 3.0.0 verfügbar (Stichwort Flexray), siehe [Release Information Device Toolbox 3.0.0](#)

Spätestens nächste Woche wird zudem die Device Toolbox 3.1.0 released werden, welche dann auch ohne Patch verwendet werden kann.

Bitte gib uns ein Feedback ob dies dein Problem bereits löst.

Danke und Gruß

Matthias

#6 - 2019-03-05 12:33 - hidden

- Description updated

- Product Issue Numbers set to <https://www.cip.audi.de/jira/projects/CDIFHD/issues/CDIFHD-24>

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

Hallo Christian,

noch ein kurzer Nachtrag:

1) Ja, du solltest auf die neuen Versionen gehen, wie in [#6357#note-5](#) empfohlen

2) Um dein eigentliches Problem zu lösen, muss das Example so erweitert werden, dass du weitere adtffileplugins nachladen kannst (darüber kannst du Deserializer bekannt machen). Dazu gibt es auch ein Ticket (CDIFHD-24). Mit der Device kannst du die typen aus ADTF 2.x bekannt machen, du müsstest das Example in etwa so erweitern:

```
// ...

#include <map>

// initialize ADTF File and Plugin Mechanism
static adtf_file::Objects oObjects;
static adtf_file::PluginInitializer oInitializer([]
{
    adtf_file::add_standard_objects();
});

void query_file_info(const std::string& filename)
{
    using namespace adtf_file;

    // load deserializer from adtffileplugin
    adtf_file::load_plugin("<path_to_device_toolbox_3_0_0>/bin/adtf_devtb_2_deserializer.adtffileplugin");
    // or debug: adtf_file::load_plugin("<path_to_device_toolbox_3_0_0>/bin/debug/adtf_devtb_2_deserializerd.adtffileplugin");

    // open file -> create reader from former added settings
    adtf_file::Reader reader(filename,
        adtf_file::getFactories<adtf_file::StreamTypeDeserializers,
            adtf_file::StreamTypeDeserializer>(),
        adtf_file::getFactories<adtf_file::SampleDeserializerFactories,
            adtf_file::SampleDeserializerFactory>(),
        std::make_shared<adtf_file::sample_factory<adtf_file::DefaultSample>>(),
        std::make_shared<adtf_file::stream_type_factory<adtf_file::DefaultStreamType>>());

    std::cout << "File: " << filename << std::endl;

    // ...

void access_file_data(const std::string& filename, const std::string& csv_file_path)
{
```

```

using namespace adtf_file;

// load deserializer from adtffileplugin
adtf_file::load_plugin("<path_to_device_toolbox_3_0_0>/bin/adtf_devtb_2_deserializer.adtffileplugin");
// or debug: adtf_file::load_plugin("<path_to_device_toolbox_3_0_0>/bin/debug/adtf_devtb_2_deserialized.adtffileplugin");

adtf_file::Reader reader(filename,
    adtf_file::getFactories<adtf_file::StreamTypeDeserializers,
        adtf_file::StreamTypeDeserializer>(),
    adtf_file::getFactories<adtf_file::SampleDeserializerFactories,
        adtf_file::SampleDeserializerFactory>(),
    std::make_shared<adtf_file::sample_factory<adtf_file::DefaultSample>>(),
    std::make_shared<adtf_file::stream_type_factory<adtf_file::DefaultStreamType>>());

StreamsInfo stream_info(reader);

```

PS: Patch an dieser Stelle nicht vergessen (siehe Device Toolbox Download), sofern du ADTF File Library 0.4.0 (= Basis ab ADTF 3.4.0) verwendest (was ich dir raten würde)

#7 - 2019-03-08 12:53 - hidden

Hallo Christian,

wir haben noch kein Feedback erhalten.
Bitte um Feedback spätestens bis zum 12.03.19
Sonst würden wir das Ticket schließen.

#8 - 2019-03-08 13:19 - hidden

Matthias Fick-Gredler wrote:

Hallo Christian,

wir haben noch kein Feedback erhalten.
Bitte um Feedback spätestens bis zum 12.03.19
Sonst würden wir das Ticket schließen.

Hallo Matthias,

werde diese Woche leider nicht mehr dazu kommen. Hab mir das aber für kommenden Montag vorgenommen und werde mich entsprechend bei euch melden.

#9 - 2019-03-08 14:46 - hidden

Hallo Christian,

das ist kein Problem.
Solange wir wissen das Du noch etwas Zeit brauchst und das Ticket noch aktuell bist, können wir das Ticket natürlich offen lassen.

Gruß
Matthias

#10 - 2019-03-12 10:00 - hidden

Hallo Matthias,

diese Woche sind bei mir leider ein paar neue Themen aufgekommen deshalb komme ich erst gegen Ende der Woche dazu mir das nochmal anzuschauen.

Viele Grüße
Christian

#11 - 2019-03-12 10:26 - hidden

Hallo Christian,

wir haben das Due Date entsprechend angepasst,
bitte bis 15.03.2019 Feedback geben.

Gruß,

Michael

#13 - 2019-03-15 12:30 - hidden

Hallo Michael,

ich konnte endlich eure Tipps ausprobieren.
Ich kann jetzt mit dem Beispielcode erfolgreich .dat Files einlesen. []
Danke für die Unterstützung!

Viele Grüße und schönes Wochenende
Christian

#14 - 2019-03-15 13:02 - hidden

- Description updated
- Status changed from Customer Feedback Required to To Be Closed
- Resolution set to Solved Issue

#15 - 2019-03-15 13:03 - hidden

- Description updated

#16 - 2019-08-08 18:19 - hidden

- Project changed from 11 to Public Support
- Status changed from To Be Closed to Closed
- Private changed from Yes to No

Files

image001.png	4.45 KB	2019-03-01	hidden
--------------	---------	------------	--------