

Public Support - Support Request #9832

adtf/anonymous would not be deserialized

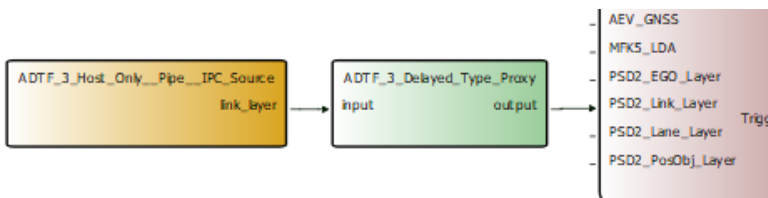
2020-01-15 09:00 - hidden

Status:	Closed	Product Issue Numbers:	https://www.cip.audi.de/jira/browse/ASU-PTB-63
Priority:	Normal	Affected Products:	ADTF 3 Support Toolbox 3.1.0, ADTF 3.6.2
Category:		Platform:	Windows 10 64bit
Customer:	AUDI	Topic:	ADTF::Common
Department:	EF	FAQ Links:	
Requester's Priority:	Normal		
Support Level:	3rd Level		
Resolution:	Product Issue Opened		

Description

Supportanfrage

Beim Versuch, Daten zwischen ADTF2 und ADTF3 auszutauschen, bekomme ich das mit der Rückrichtung noch nicht hin (3 -> 2). Inhalt soll einfach ein beliebig großer Byte-Blob sein.

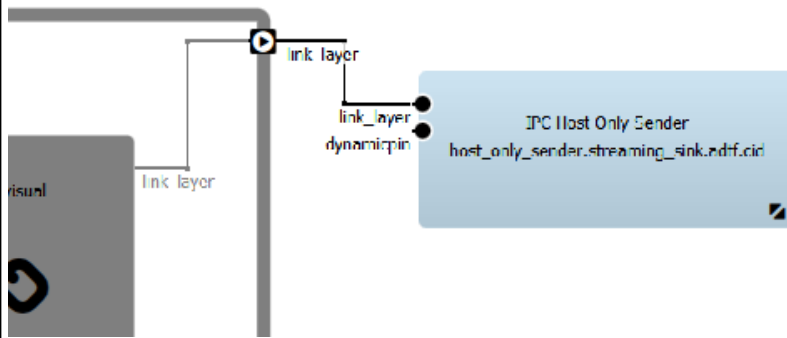


Bekomme in ADTF2 die Fehlermeldung:

```
error | 00:05:39:390 | Unable to read from input stream: Class not found: Class not found:  
"oid.adtf.adtf3.type_deserialization.adtf/anonymous" | ipc_sample_source.cpp(157) | 16692/23548 |  
E:\CONAN\conan\data\ADTF\2.14.0\ex24\testing\package\85f780d0530411a64b0be4407b381706014b445d\bin\adtf_launcher.  
exe | OK | No error | cIPCPlugin
```

Hier der Code-Snippet aus dem ADTF3-Sender (ich habe leider kein Beispiel gefunden, wo ein Blob geschickt wird)

```
writerLinkLayer_ = CreateOutputPin(&quot;link_layer&quot;;, adtf::streaming::  
stream_meta_type_anonymous());  
  
// ...  
  
std::string serializedData;  
serialize_to_string(*link_layer_short_range, serializedData);  
adtf::ucom::object_ptr_locked<adtf::streaming::ISampleBuffer> pBuffer;  
adtf::ucom::object_ptr<adtf::streaming::ISample> pSample;  
adtf::streaming::alloc_sample(pSample, tTimeStamp(0));  
adtf::streaming::write_to_sample(*pSample, serializedData);  
writerLinkLayer_->Write(pSample);
```



Lösung

Dafür gibt's in der Toolbox tatsächlich keinen Deserialisierer. Im Moment fällt mir für Dich als Workaround nur ein einen adtf/default Type zu verwenden, der einfach eine leere Struktur beschreibt. Dann sollte es klappen.

Ticket ASUPTB-63 in der Support Toolbox erstellt, damit es da in Zukunft einen Deserialisierer gibt.

Was mir in deinem Code noch aufgefallen ist:

```
std::string serializedData;
serialize_to_string(*link_layer_short_range, serializedData);
adtf::streaming::alloc_sample(pSample, tTimeStamp(0));
adtf::streaming::write_to_sample(*pSample, serializedData);
```

write_to_sample funktioniert nicht mit std::string, sondern nur mit std::is_trivially_copyable Daten Typen (das fehlt leider im template als static_assert).

mach am besten folgendes:

```
std::string serializedData;
serialize_to_string(*link_layer_short_range, serializedData);
adtf::ucom::object_ptr<adtf::streaming::ISample> pSample;
adtf::streaming::alloc_sample(pSample, tTimeStamp(0));

{
    adtf::ucom::object_ptr_locked<adtf::streaming::ISampleBuffer> pBuffer;
    RETURN_IF_FAILED(pSample->WriteLock(pBuffer));
    RETURN_IF_FAILED(pBuffer->Write(adtf::base::adtf_memory_buffer<const char>(serializedData.data
    ()), serializedData.length()));
}
```

History

#1 - 2020-01-15 09:02 - hidden

- Subject changed from Re: AW: AW: ADTF3 Hack&Ack Offene Fragen to ADTF3 Hack&Ack Offene Fragen
- Status changed from New to In Progress
- Author changed from hidden to hidden
- Topic set to ADTF::Common
- Customer set to AUDI
- Department set to EF

#2 - 2020-01-15 09:03 - hidden

- Description updated

#3 - 2020-01-15 09:04 - hidden

- Description updated

#4 - 2020-01-15 10:33 - hidden

Hi Falk,

uiuiui, dafür gibt's in der Toolbox tatsächlich keinen Deserialisierer. Im Moment fällt mir für Dich als Workaround nur ein einen adtf/default Type zu

verwenden, der einfach eine leere Struktur beschreibt. Dann sollte es klappen.

Ich erstell ein Ticket in der Support Toolbox, damit es da in Zukunft einen Deseiralisierer gibt.

Was mir in deinem Code noch Aufgefallen ist:

```
std::string serializedData;  
serialize_to_string(*link_layer_short_range, serializedData);  
adtf::streaming::alloc_sample(pSample,tTimeStamp(0));  
adtf::streaming::write_to_sample(*pSample, serializedData);
```

write_to_sample funktioniert nicht mit std::string, sondern nur mit std::is_trivially_copyable Daten Typen (das fehlt leider im template als static_assert).

mach am besten folgendes:

```
std::string serializedData;  
serialize_to_string(*link_layer_short_range, serializedData);  
adtf::ucom::object_ptr<adtf::streaming::ISample> pSample;  
adtf::streaming::alloc_sample(pSample,tTimeStamp(0));  
  
{  
    adtf::ucom::object_ptr_locked<adtf::streaming::ISampleBuffer> pBuffer;  
    RETURN_IF_FAILED(pSample->WriteLock(pBuffer));  
    RETURN_IF_FAILED(pBuffer->Write(adtf::base::adtf_memory_buffer<const char>(serializedData.data(), serializ  
edData.length()));  
}
```

Grüße,

Martin

#5 - 2020-01-15 10:36 - hidden

- Resolution set to Product Issue Opened
- Product Issue Numbers set to <https://www.cip.audi.de/jira/browse/ASUPTB-63>

#6 - 2020-01-15 14:23 - hidden

- Status changed from In Progress to Customer Feedback Required
- Support Level changed from 2nd Level to 3rd Level

#7 - 2020-01-15 14:23 - hidden

- Project changed from Public Support to 11
- Affected Products ADTF 3.6.2 added
- Platform Windows 10 64bit added

#8 - 2020-01-15 14:24 - hidden

- Subject changed from ADTF3 Hack&Ack Offene Fragen to adtf/anonymous would not be deserialized
- Affected Products ADTF 3 Support Toolbox 3.1.0 added

#9 - 2020-01-16 13:07 - hidden

Hallo Martin,

ich habe den stream_meta_type_anonymous() zu stream_meta_type_plain() geändert und deine write-methode übernommen. Damit kommen nun Daten im ADTF2 an.

Vielen Dank, kann geschlossen werden.

#10 - 2020-01-16 14:51 - hidden

- Status changed from Customer Feedback Required to To Be Closed

#11 - 2020-01-16 15:05 - hidden

- Description updated

#14 - 2020-07-07 13:27 - hidden

- Project changed from 11 to Public Support

- Private changed from Yes to No

#15 - 2020-07-07 16:40 - hidden

- Status changed from To Be Closed to Closed

Files

image001.png	11.4 KB	2020-01-15	hidden
image002.png	9.36 KB	2020-01-15	hidden