Public Support - Support Request #3349

How to access Media Description in ADTF 3

2018-07-16 09:24 - hidden

Status:	Closed		
Priority:	Normal		
Category:			
Customer:	AUDI	**Product Issue Numbers:**	
Department:	AEV	**Affected Products:**	ADTF 3.3.1
Requester's Priority:	Normal	**Platform:**	
Support Level:	2nd Level	**Topic:**	ADTF::MediaDescription
Resolution:	Solved Issue	**FAQ Links:**	

Description

Supportanfrage

Ich versuche gerade Filtercode von ADTF2
nach ADTF3 zu portieren. Früher war folgendes in ADTF2 möglich:

```
cObjectPtr<adtf::IMediaDescriptionManager> pDescManager;
RETURN_IF_FAILED(_runtime->GetObject(OID_ADTF_MEDIA_DESCRIPTION_MANAGER,
                                     IID_ADTF_MEDIA_DESCRIPTION_MANAGER,
                                     (tVoid**)&pDescManager,
                                     __exception_ptr));
cString strDesc = pDescManager->GetMediaDescription("tSimple");
```

In ADTF3 sollte das so ähnlich gehen:

```
object_ptr<IStreamType> pTypeIn;
object_ptr<adtf::services::IMediaDescriptionService> pDescManager;
cString strDesc;

if (IS_FAILED(_runtime->GetObject(pDescManager, "ADTF Media Description Service")))
{
    // COULD NOT INSTANTIATE DESCRIPTION MANAGER
}

// GET SIGNAL DESCRIPTION STRING FROM *.DESCRIPTION FILE (VIA ADTF'S DESCRIPTION MANAGER)
pDescManager->GetStructMediaDescription("tSimple", strDesc);

// CREATE STREAM TYPE BASED ON DESC
create_adtf_default_stream_type("tSimple",
                                strDesc,
                                pTypeIn,
                                m_oCodecFactory);
```

cString zu verwenden funktioniert an dieser Stelle nur leider nicht mehr, denn er erwartet eine initialisierbare Klasse,
die sich vom Interface IString ableitet. Welche Stringimplementierung kann ich da am besten nehmen?
Ich hoffe außerdem, dass ich die GetObject Methode mit dem richtigen String für den Media Description Manager aufrufe,
weil OID_ADTF_MEDIA_DESCRIPTION_MANAGER gibt es ja offensichtlich in ADTF3 nicht mehr.

Lösung

Über das ADTF Media Description Package, im Data Defintion Language Package Media Description Package ist dann speziell der Zugriff beschrieben (siehe Accessing Data with a Decoder/Codec).

Das Snippet zielt auf die allgemeinere cCodec Variante, hier wird auch nicht create_adtf_default_stream_type verwendet.
Dieser benötigt eine die spezialisierte Form cSampleCodec.
Dementsprechend sind auch die Factories gestaffelt, siehe Hierarchie

Wenn du also create_adtf_default_stream_type verwendest, dann musst du die Snippets ebenso spezialisieren.
Das machen wir in den Beispielen, daran kannst du dich orientieren.

History

#1 - 2018-07-16 09:25 - hidden

- Status changed from New to In Progress

- Author changed from hidden to hidden

#3 - 2018-07-16 09:31 - hidden

- Status changed from In Progress to Customer Feedback Required

Hallo Christoph,

Über das ADTF Media Description Package, im Data Defintion Language Package Media Description Package ist dann speziell der Zugriff beschrieben (siehe Accessing Data with a Decoder/Codec)
Desweiteren kannst du dich an unseren Standardbsp. orientieren, z.B. Simple Data Triggered Filter (.\src\examples\src\adtf\filters\standard_filters\data_triggered_filter\)

Hilft dir das weiter ?

#4 - 2018-07-16 10:09 - hidden

Hallo Florian,

das hilft mir leider nicht weiter. Ich orientiere mich bereits am Standardbeispiel.
Im Standardbeispiel wird die Klasse cSampleCodecFactory verwendet und die Funktion Register(), um den Typ der Eingangspins festzulegen.

Aber okay, nehmen wir mal an, ich würde nun die Klasse cCodecFactory anstelle von cSampleCodecFactory verwenden,
wie im Link unter "Accessing Data with a Decoder/Codec" beschrieben. Dann stellen sich mir zwei Fragen:

- wie soll ich dann meine Eingangspins registrieren?
  Soweit ich das richtig sehe, arbeitet die Funktion create_adtf_default_stream_type() mit dem Typ cSampleCodecFactory,
  nicht mit cCodecFactory.
- was genau ist der Unterschied zwischen beiden Klassen cCodecFactory und cSampleCodecFactory (geht aus der Doku nicht wirklich hervor)

#5 - 2018-07-16 12:50 - hidden

Hallo Christoph,

ja, das Snippet zielt auf die allgemeinere cCodec Variante, hier wird auch nicht create_adtf_default_stream_type verwendet.
Dieser benötigt eine die spezialisierte Form cSampleCodec.
Dementsprechend sind auch die Factories gestaffelt, siehe Hierarchie

Wenn du also create_adtf_default_stream_type verwendest, dann musst du die Snippets ebenso spezialisieren.
Das machen wir in den Beispielen, daran kannst du dich orientieren.

In ADTF 3.x ist es wichtig, den Use Case rauszustellen, eine einfache Portierung von ADTF 2.x klappt nicht immer, da es auch unterschiedliche Anforderungen und auch Architektur gibt.

#6 - 2018-07-23 10:55 - hidden

Hallo Florian,

vielen herzlichen Dank für Deinen Support!
Es funktioniert jetzt.

Viele Grüße
Christoph

#7 - 2018-07-25 09:09 - hidden

- Project changed from 11 to Public Support

- Description updated

- Status changed from Customer Feedback Required to To Be Closed

- Private changed from Yes to No

- Resolution set to Solved Issue

#8 - 2018-07-31 15:29 - hidden

- Status changed from To Be Closed to Closed