

Public Support - Support Request #2764

MSF_REFERENCE

2018-05-28 15:45 - hidden

Status: Closed	
Priority: Normal	
Category:	
Customer: AUDI	Product Issue Numbers:
Department: EX	Affected Products: ADTF 2.14.2, ADTF 3.3.0
Requester's Priority: Normal	Platform: Windows 7 64bit
Support Level: 2nd Level	Topic: ADTF::SDK
Resolution: Workaround Available	FAQ Links:
Description	
Supportanfrage:	
Hallo,	
Jan möchte Kopiervorgänge sparen und externen Speicher in einem MediaSample referenzieren. Uns ist bewusst, dass das seine Gefahren birgt, aber wir wollen es dennoch versuchen ;)	
Problem ist allerdings, dass wir von außen nicht mitbekommen, wann der Referenz-Counter des MediaSamples auf 0 geht und somit das MediaSample zerstört wird. Das wäre der Zeitpunkt, an dem wir auch den externen Speicher recyceln können.	
Idee für einen Workaround:	
· Eigene Implementierung eines MediaSamples, welches einen CallBack anbietet. Problem: Neue Klasse müsste in der Runtime erst registriert werden.	
Habt ihr eine Alternative zu unserem Workaround?	
Lösung:	
Auszug aus der ADTF3 Doku:	
<pre>/** * Sample that uses samples buffers that do not copy the data, but only keep a reference to it. * Use this with EXTREME care!! Keep these samples in a * \ref adtf::ucom::ant::weak_object_ptr in order to be able to check if they no longer exist * before you manipulate the data again. */</pre>	
class cReferenceSample: public cSample	
Dort gibt es das Beispiel cDemoSampleDataReferenceSource, mit einem weak_object_ptr kann überprüft werden, ob das Sample noch in Verwendung ist.	
zu ADTF 2:	
Mangels weak_object_ptr in ADTF2 würde man nicht um eine eigene Implementierung eines MediaSamples herumkommen	
Workaround: Für speziellen Anwendungsfall hinreichend großen Ringspeicher nutzen, um sicherzustellen, dass überschriebener Speicher nicht mehr genutzt wird.	

History

#1 - 2018-05-29 09:41 - hidden

- Status changed from New to Customer Feedback Required

- Topic set to ADTF::SDK

- Customer set to AUDI

Hallo Jens,

um welche ADTF Version handelt es sich in diesem Fall?

Gruß,
Michael

#3 - 2018-05-29 13:48 - hidden

- Author changed from hidden to hidden

#4 - 2018-05-29 14:22 - hidden

Hier der Auszug aus der ADTF3 Doku:

```
/**
 * Sample that uses samples buffers that do not copy the data, but only keep a reference to it.
 * Use this with EXTREME care!! Keep these samples in a
 * \ref adtf::ucom::ant::weak_object_ptr in order to be able to check if they no longer exist
 * before you manipulate the data again.
 */
class cReferenceSample: public cSample
```

Da gibts auch das Beispiel cDemoSampleDataReferenceSource. Dort sieht man, dass man mit einem weak_object_ptr überprüfen kann, ob das Sample noch in Verwendung ist.

Wenn das nicht reicht hilft wirklich nur eine eigene Implementierung die im Destruktor ein Callback ruft. Eine neue Media Sample Klasse musst man nicht in der Runtime registrieren, so wie in dem Beispiel zu sehen. Das war in ADTF2 Zeiten nur wegen der Serialisierung notwendig, die ist in ADTF3 aber vom Sample losgelöst.

#5 - 2018-05-29 14:55 - hidden

- Author changed from hidden to hidden

#6 - 2018-05-29 14:57 - hidden

Hallo Jens,

sieh dir bitte mal die Antwort von Martin Heimlich an.

Gruß,
Michael

#7 - 2018-05-29 17:47 - hidden

Hallo,

meine Anfrage bezieht sich auf ADTF2.

Ich nehme mit, dass wir in ADTF3 einen praktikablen Weg haben.

Mangels weak_object_ptr in ADTF2 würde ich aber in ADTF2 nicht um eine eigene Implementierung eines MediaSamples herumkommen, richtig?

Workaround: Für speziellen Anwendungsfall hinreichend großen Ringspeicher nutzen, um sicherzustellen, dass überschriebener Speicher nicht mehr genutzt wird.

MfG Jens

#8 - 2018-05-30 09:15 - hidden

Ja genau.

#9 - 2018-05-30 14:12 - hidden

Danke, Ticket kann geschlossen werden.

#10 - 2018-05-30 15:06 - hidden

- Description updated

- Status changed from Customer Feedback Required to To Be Closed

- Resolution set to Workaround Available

- Department set to EX

- *Affected Products ADTF 2.14.2, ADTF 3.3.0 added*
- *Platform Windows 7 64bit added*

#12 - 2018-06-05 15:18 - hidden

- *Status changed from To Be Closed to Closed*

#13 - 2018-09-11 17:09 - hidden

- *Private changed from Yes to No*

#14 - 2020-07-07 17:17 - hidden

- *Description updated*