

ADTFS-46407 External Clock

2017-03-24 10:45 - hidden

Status:	Closed	
Priority:	Normal	
Category:		
Customer:	ELEKTROBIT	Product Issue Numbers:
Department:	SUPPORT	Affected Products: ADTF 2.12.1
Requester's Priority:	Normal	Platform: Windows 7 64bit
Support Level:	2nd Level	Topic: ADTF::Clock
Resolution:	Solved Issue	FAQ Links:

Description

Supportanfrage:

Hintergrund:
"Demo Clock Filter" wird in einer Konfiguration mit einem HDRec verwendet. Laut Doku wird die Clock in dem DAT file "gespeichert", bzw. die tADTFClockRelation. Das DAT soll in einer neuen Konfiguration abgespielt werden, hier werden die Zeitstempel beider Uhren benötigt.

Daher:

a) Aus der Doku heraus fehlt mir die Vorstellung was mit der Clock Relation gemeint ist. Wird in dieser extension wirklich nur alle 5s so eine Relation abgespeichert? Wenn ja, sind 5s nicht ein ziemlich großer Abstand?

b) Bekommt man dann aus jedem MediaSample beide Clocks mit den entsprechenden Zeitstempeln? Oder muss man das immer "von Hand" mit der Clock Relation synchronisieren?

Viele Grüße,
Anja

Lösung:

In ADTF gibt es default mäßig zwei verschiedene Zeitgeber, die ADTF clock (ADTFTime) und die ADTF stream clock (StreamTime).

ADTFTime: Zeit seit ADTF Start
StreamTime: Zeit ab dem Zeitpunkt des RunLevels: Running.

a) Aus der Doku heraus fehlt mir die Vorstellung was mit der Clock Relation gemeint ist. Wird in dieser extension wirklich nur alle 5s so eine Relation abgespeichert? Wenn ja, sind 5s nicht ein ziemlich großer Abstand?

Die clock relation ist die Beziehung zwischen der StreamTime und dem jeweiligen Zeitgeber.
z.B. adtf_stream (StreamTime) & ADTFTime
oder für selbst implementierte Zeitgeber: StreamTime & MyClock

Da nur lediglich der Offset zwischen beiden Zeiten gespeichert wird, ist es egal das es "nur" alle ~5 Sekunden geschieht.

b) Bekommt man dann aus jedem MediaSample beide Clocks mit den entsprechenden Zeitstempeln? Oder muss man das immer "von Hand" mit der Clock Relation synchronisieren?

Jedes MediaSample bekommt bei der Aufnahme mit dem HDR alle zur Zeit der Aufnahme verfügbaren clocks.

adtf_clock_ext:

This extension contains an array of all recorded clocks stored in adtf::tADTFClockExtensionItem elements. The first element specifies the ADTFTime clock and the second element the StreamTime clock that have been used during recording. The remaining elements list all clocks. The amount of elements is defined by dividing the size of the extension by

sizeof(adtf::tADTFClockExtensionItem).

Laut SDK doku werden alle clock extensions die aufgenommen wurden in einem array aus adtf::tADTFClockExtensionItem elementen gespeichert.

Ich hab leider kein code Beispiel, wie man an diese heran kommt.

Der Aufbau sieht so aus:

```
typedef struct
{
    tChar strClockName[256]; // ADTF_MAX_EXTERNAL_CLOCK_NAME_LENGTH
} tADTFClockExtensionItem;
```

History

#1 - 2017-03-24 11:32 - hidden

- Topic set to ADTF::Clock
- Customer set to ELEKTROBIT
- Department set to SUPPORT

Hallo Anja,
bitte wenn möglich immer die ADTF Version, ggf. Toolbox(en) und OS mit angeben

#2 - 2017-03-24 12:15 - hidden

Danke für den Hinweis.

ADTF 2.12.1 Toolbox unbekannt. Windows 7

Gruß,
Anja

#3 - 2017-03-24 15:23 - hidden

- Status changed from New to Customer Feedback Required
- Support Level changed from 2nd Level to 3rd Level
- Affected Products ADTF 2.12.1 added
- Platform Windows 7 64bit added

An den Anwender:

Hallo Anja,

in ADTF gibt es default mäßig zwei verschiedene Zeitgeber, die ADTF clock (ADTFTime) und die ADTF stream clock (StreamTime).

ADTFTime: Zeit seit ADTF Start

StreamTime: Zeit ab dem Zeitpunkt des RunLevels: Running.

- a) Aus der Doku heraus fehlt mir die Vorstellung was mit der Clock Relation gemeint ist. Wird in dieser extension wirklich nur alle 5s so eine Relation abgespeichert? Wenn ja, sind 5s nicht ein ziemlich großer Abstand?

Die clock relation ist die Beziehung zwischen der StreamTime und dem jeweiligen Zeitgeber.

z.B. adtf_stream (StreamTime) & ADTFTime

oder für selbst implementierte Zeitgeber: StreamTime & MyClock

Da nur lediglich der Offset zwischen beiden Zeiten gespeichert wird, ist es egal das es "nur" alle ~5 Sekunden geschieht.

- b) Bekommt man dann aus jedem MediaSample beide Clocks mit den entsprechenden Zeitstempeln? Oder muss man das immer "von Hand" mit der Clock Relation synchronisieren?

Jedes MediaSample bekommt bei der Aufnahme mit dem HDR alle zur Zeit der Aufnahme verfügbaren clocks.

#4 - 2017-03-29 12:39 - hidden

Bitte um Rückmeldung bis morgen den 30.03.2017.

#5 - 2017-03-29 13:30 - hidden

- File *exportSink.cpp* added

- File *exportSink.hpp* added

Hallo Stephan,

Jedes MediaSample bekommt bei der Aufnahme mit dem HDR alle zur Zeit der Aufnahme verfügbaren clocks.

Kannst du mir das genauer erklären? Wie kommt man an die clocks im MediaSample ran? Hast du da ein Code Beispiel?

Fabian hat ein Beispiel geschrieben das die Zeiten ausrechnet. Ist sich aber auch nicht ganz sicher. Kann man das so machen? Der wichtige Teil dabei ist in *exportSink.cpp* (Verzeichnis *importExportService*) die Methode *RegisterExtensions*.

Best regards,
Anja Winkler
EB Assist ADTF Support-Team

#6 - 2017-03-30 16:57 - hidden

An den Anwender:

Hallo Anja,

adtf_clock_ext:

This extension contains an array of all recorded clocks stored in *adtf::tADTFClockExtensionItem* elements. The first element specifies the ADTFTIME clock and the second element the StreamTime clock that have been used during recording. The remaining elements list all clocks. The amount of elements is defined by dividing the size of the extension by *sizeof(adtf::tADTFClockExtensionItem)*.

Laut SDK doku werden alle clock extensions die aufgenommen wurden in einem array aus *adtf::tADTFClockExtensionItem* elementen gespeichert. Ich hab leider kein code Beispiel, wie man an diese heran kommt.

Der Aufbau sieht so aus:

```
typedef struct
{
    tChar strClockName[256]; // ADTF_MAX_EXTERNAL_CLOCK_NAME_LENGTH
} tADTFClockExtensionItem;
```

Hilft euch das weiter?

#7 - 2017-03-31 15:46 - hidden

Zeit für Feedback bis zum 04.04.2017 eingeräumt.

#8 - 2017-04-06 15:07 - hidden

- Description updated

- Status changed from Customer Feedback Required to To Be Closed

- Resolution set to Solved Issue

Hallo Anja,

falls wir das Ticket noch public setzten dürfen sagt uns Bescheid.

@Florian, Ticket kann geschlossen werden.

#9 - 2017-04-07 14:15 - hidden

- Status changed from To Be Closed to Closed

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

#10 - 2017-04-07 14:15 - hidden

- Project changed from Public Support to 7

#11 - 2018-01-25 12:14 - hidden

- Project changed from 7 to Public Support

#12 - 2018-01-25 12:16 - hidden

- Private changed from Yes to No

Files

exportSink.cpp	1.77 KB	2017-03-29	hidden
exportSink.hpp	1000 Bytes	2017-03-29	hidden