

Public Support - Support Request #16382

Clock handling for timer runner and playback

2022-01-14 11:15 - hidden

Status:	Closed	Product Issue Numbers: Affected Products: ADTF 3.13.2 Platform: Windows 10 64bit Topic: ADTF::Clock FAQ Links:
Priority:	Normal	
Category:		
Customer:	VW	
Department:	IAV	
Requester's Priority:	Normal	
Support Level:	2nd Level	
Resolution:	Solved Issue	
Description		
Supportanfrage		
<p>Ich würde gerne wissen, auf welcher Zeitbasis der „Default Timer Runner“ läuft.</p> <p>https://support.digitalwerk.net/adtf/v3/adtf_html/page_default_core_objects_plugin.html#section_default_timer_runner</p> <p>Ist das die „Stream Time“?</p>		
Lösung		
<p>Sämtliches Timing in ADTF wird von der Reference Clock übernommen, zur Laufzeit spricht man von der Stream Time, unabhängig davon, wer sich angemeldet hat, der Taktgeber zu sein für die Reference Clock zu sein.</p> <p>Siehe auch https://support.digitalwerk.net/adtf/v3/adtf_html/page_clock_concept.html</p> <p>Im Playback schreitet die Zeit diskret voran. Die Timer holen allerdings "verpasste" Takte geblockt beim ersten Sample nach der nächsten Deadline nach.</p> <p>Das wird sich in einer der nächsten Versionen von ADTF verbessern, in dem die Timer die Möglichkeit bekommen über die Clock dem Player (und seiner Clock) mitzuteilen wann sie denn das nächste mal geweckt werden wollen.</p> <p>Zusätzlich wurde im Player noch die "reset_recording_offset" Property im Vgl. zu ADTF 2.x entfernt. Heißt die Sample Timestamps und die Streamtime (basierend auf der Chunktime) werden nie mehr manipuliert und sind exakt die Selben wie bei der Aufzeichnung.</p>		

History

#1 - 2022-01-14 13:34 - hidden

- Project changed from Public Support to 20
- Description updated
- Status changed from New to Customer Feedback Required
- Customer set to VW
- Department set to IAV
- Topic set to ADTF::Clock

Hallo Sebastian,

sämtliches Timing in ADTF wird von der Reference Clock übernommen, zur Laufzeit spricht man von der Stream Time, unabhängig davon, wer sich angemeldet hat, der Taktgeber zu sein für die Reference Clock zu sein.
Siehe auch https://support.digitalwerk.net/adtf/v3/adtf_html/page_clock_concept.html

#2 - 2022-01-19 13:15 - hidden

Danke für die Antwort. Also hier nochmal um mein Verständnis zu verifizieren:

Im Playback Mode wird, solange man nicht eine eigene clock registriert, die Reference Clock durch die timestamps der Samples im DAT-File erzeugt. Solange man nicht die Interpolation im Playback Service aktiviert hat, schreitet die Reference Clock also diskret mit den Zeitstempeln aus den samples fort.
Der „Default Timer Runner“ hat damit im Playback Mode maximal die Auflösung der Zeitstempel aus dem DAT file.

Wenn ich also z.b. einen Timer Runner habe, der mit 10ms triggern soll, aber alle Daten im DAT-File nur alle 100ms aufgezeichnet wurden, wird der runner nie diese Zykluszeit erreichen.

Mir ist klar das dieses Beispiel etwas konstruiert ist, weil man ja oft viele streams in einem DAT-File hat und die Reference Clock über alle samples hinweg gebildet wird, sodass dieses Szenario meist nicht auftreten kann.

#3 - 2022-01-19 19:42 - hidden

- Status changed from Customer Feedback Required to In Progress

#4 - 2022-01-20 06:52 - hidden

Hi Sebastian,

ja genau, im Playback schreitet die Zeit diskret voran. Die Timer holen allerdings "verpasste" Takte geblockt beim ersten Sample nach der nächsten Deadline nach.

Das wird sich in einer der nächsten Versionen von ADTF verbessern, in dem die Timer die Möglichkeit bekommen über die Clock dem Player (und seiner Clock) mitzuteilen wann sie denn das nächste mal geweckt werden wollen.

Grüße,

Martin

#5 - 2022-01-20 07:42 - hidden

- Status changed from In Progress to Customer Feedback Required

#6 - 2022-01-20 12:12 - hidden

Hi Martin,

das bringt mich noch auf eine andere Frage. Ist die Bildung der timestamps in recorder und player noch prinzipiell so, wie bei ADTF2?:

https://support.digitalwerk.net/adtf/v2/adtf_sdk_html_docs/page_add_adtf_times.html

Also werden die sampletimes nicht im recorder angefasst, sondern zusätzlich eine chunktime gespeichert, die dann als Zeitquelle für den playback service dient?

Grüße

Sebastian

#7 - 2022-01-20 12:24 - hidden

Ja genau!

Zusätzlich wurde im Player noch die "reset_recording_offset" Property entfernt. Heißt die Sample Timestamps und die Streamtime (basierend auf der Chunktime) werden nie mehr manipuliert und sind exakt die Selben wie bei der Aufzeichnung.

Grüße,

Martin

#8 - 2022-01-20 12:28 - hidden

Ok. Dann vielen Dank für den super Support!

Von mir aus kann das Ticket jetzt geschlossen werden.

Grüße

Sebastian

#9 - 2022-01-20 12:38 - hidden

- Status changed from Customer Feedback Required to To Be Closed

- Resolution set to Solved Issue

- Affected Products ADTF 3.13.2 added

- Platform Windows 10 64bit added

Immer gerne!

#10 - 2022-01-20 12:52 - hidden

- Project changed from 20 to Public Support

- Subject changed from Frage zu timerrunner to Clock handling for timer runner and playback

- Description updated

- *Private changed from Yes to No*

#11 - 2023-01-17 16:13 - hidden

- *Status changed from To Be Closed to Closed*