

Public Support - Support Request #10847

Notation of timestamps for the adtf_datexporter.exe

2020-03-19 11:30 - hidden

Status:	Closed	
Priority:	Normal	
Category:		
Customer:	VW	Product Issue Numbers:
Department:	IAV	Affected Products: ADTF 2.13.2
Requester's Priority:	Normal	Platform: Windows 7 64bit
Support Level:	2nd Level	Topic: ADTF::Datexporter
Resolution:	Solved Issue	FAQ Links:

Description

Support Anfrage:

ich habe eine Frage zur Verwendung des adtf_datexporter für ADTF 2.13.2.

Ich möchte aus einem bestehenden dat File, mit einem CAN Datenstrom welches 60s lang ist, den CAN Datenstrom im Streaming Zeitbereich von 10s-20s als neues dat File extrahieren.

Hierfür verwende ich die Cut and import Funktion des Exporters mit der Syntax aus der Hilfe:

adtf_datexporter.exe

```
-import <path to DAT file to import from>.dat  
-stream <stream1> -start <stream1_start> -end <stream1_end>  
-stream <stream2> -start <stream2_start> -end <stream2_end>  
-ouput <path to output file>.dat
```

Als Beispiel:

```
adtf_datexporter.exe  
-import "my_can_dat.dat"  
-stream CAN -start ? -end ?  
-ouput "my_cut_can_dat.adt"
```

Oder

```
adtf_datexporter.exe  
-import "my_can_dat.dat"  
-start ? -end ?  
-stream CAN  
-ouput "my_cut_can_dat.adt"
```

Wie muss die Notation der Zeitangaben für -start und -end aussehen, wenn ich den angegebenen Zeitbereich von 10s-20s extrahieren möchte?

Welche der beiden Varianten kann verwendet werden?

Danke schonmal.

Viele Grüße

Steve

Lösung:

Die Streams müssen zuerst spezifiziert werden, dann die timestamps hinterher, heißt dein erstes Beispiel wäre hier richtig:

```
-stream x -start [timestamp] -end [timestamp]
```

Einzufügen sind die Timestamps, die bei deinen angepeilten Zeitmarken liegen.

Über Tools > Show Dat File Info in ADTF kannst du dir die Samples mit den zugehörigen Timestamps anzeigen lassen und die, die du brauchst herausfinden

Bei der Einheit um Mikrosekunden.

ADTF 2.x ist jeder tTimestamp in us Auflösung.

In ADTF 3.x ebenso, wobei es hier mittlerweile durch tNanoSeconds sogar überholt ist und präziser auflösen kann.

Es gibt aber hier Funktionen für beide Zeitschnitte.
Das aber nur als Ausblick, ADTF 2.x ist und bleibt Microseconds.

Mit 2.14.2 getestet und dem example file darin:

```
c:\tools\adtf\2.14.2\bin>adtf_datexporter.exe -liststreams templates\example_test_file.dat
```

```
Streams in c:\tools\adtf\2.14.2\bin\templates\example_test_file.dat
```

```
Source: adtf.import.dat_file_source
```

```
  Name      Start    End
```

```
-----
```

```
video      113      14798822
```

```
  available sinks:
```

```
    BMP Video Sink: adtf.export.bmp
```

```
    CSV Sink: adtf.export.csv
```

```
NESTED_STRUCT      113      14805419
```

```
  available sinks:
```

```
    CSV Sink: adtf.export.csv
```

```
c:\tools\adtf\2.14.2\bin>adtf_datexporter.exe -import templates\example_test_file.dat -stream video new_video_dat_full.dat
```

```
..\..\..\..\source\src\services\datexporter\datexporter_srv.cpp(163): info 0x00000000: Processed 0%
```

```
..\..\..\..\source\src\services\datexporter\datexporter_srv.cpp(163): info 0x00000000: Processed 100%
```

```
c:\tools\adtf\2.14.2\bin>adtf_datexporter.exe -import templates\example_test_file.dat -stream video -start 5000000 -end 10000000 new_video_dat_5s_to_10s.dat
```

```
..\..\..\..\source\src\services\datexporter\datexporter_srv.cpp(163): info 0x00000000: Processed 34%
```

Im org. Aufruf war auch noch ein Typo vorhanden (-ouput).

Problem wurde behoben

History

#1 - 2020-03-19 13:06 - hidden

- Project changed from Public Support to 20

- Status changed from New to In Progress

- Topic set to ADTF::Datexporter

- Customer set to VW

- Department set to IAV

- Affected Products ADTF 2.13.2 added

#2 - 2020-03-19 14:16 - hidden

- Status changed from In Progress to Customer Feedback Required

Hallo,

die Streams müssen zuerst spezifiziert werden, dann die timestamps hinterher, heißt dein erstes Beispiel wäre hier richtig:

```
-stream x -start [timestamp] -end [timestamp]
```

Einzufügen sind die Timestamps, die bei deinen angepeilten Zeitmarken liegen.

Über Tools > Show Dat File Info in ADTF kannst du dir die Samples mit den zugehörigen Timestamps anzeigen lassen und die, die du brauchst herausfinden

Hilft dir das weiter?

Gruß

Michael

#3 - 2020-03-19 14:45 - hidden

Hallo Michael,

leider hilft mir die Antwort nur bedingt weiter.

Ob ich jetzt im Zeitraum von 10s-20s oder von 5s-6s herauschneiden möchte ist erst einmal flexibel zu betrachten. Mir geht es hierbei um die Notation des Feldes [timestamp].

- Ist es eine Zahl für die Streamingtime in Sekunden (-start 5 -end 6) oder (-start 5.0 -end 6.0)?
- Ist es eine Zahl für die Streamingtime in Millisekunden (-start 5000 -end 6000) oder in Microsekunden oder gar Nanosekundenangabe?
- Oder ist es eine Notation der Zeitangabe in hh:mm:ss.ms (-start 00:00:05.000 -end 00:00:06.000)

Viele Grüße
Steve

#4 - 2020-03-19 15:38 - hidden

- Status changed from Customer Feedback Required to In Progress

Hallo Steve,

Es handelt sich bei der Einheit um Mikrosekunden

Gruß
Michael

#5 - 2020-03-19 15:39 - hidden

- Status changed from In Progress to Customer Feedback Required

#6 - 2020-03-25 09:19 - hidden

Kann das Ticket damit geschlossen werden?
Bitte Feedback geben bis 27.03.2020

#7 - 2020-03-25 11:00 - hidden

Hallo,

nein kann es noch nicht, da ich noch auf Rückantwort zu meiner Mail vom 19.03.2020 14:37 warte.

Die Antwort Millisekunden ist leider nicht ausreichend, da der Aufruf für einen Schnitt des CAN im Bereich von 5s-6s nicht funktionsfähig ist:

```
ADTF\bin\adtf_datexporter.exe ^
  -import &quot;F:\my_dat.dat&quot; ^
  -stream RawCanFD -start 5000 -end 6000 ^
  -output &quot;F:\my_dat_can.dat&quot;
```

#8 - 2020-03-25 11:03 - hidden

Hallo Steve,

Die Antwort Millisekunden ist leider nicht ausreichend, da der Aufruf für einen Schnitt des CAN im Bereich von 5s-6s nicht funktionsfähig ist:

Die Antwort waren allerdings **Mikrosekunden**, nicht Milli.
ADTF 2.x ist jeder tTimestamp in us Auflösung.

In ADTF 3.x ebenso, wobei es hier mittlerweile durch tNanoSeconds sogar überholt ist und präziser auflösen kann.
Es gibt aber hier Funktionen für beide Zeitschnitte.

Das aber nur als Ausblick, ADTF 2.x ist und bleibt Microseconds.

#9 - 2020-03-25 11:30 - hidden

Hallo,

auch die Änderung auf Mikrosekunden bringt leider keinen Erfolg.
Der Exportprozess wird auch nicht in der Kommandozeile angezeigt.

```
ADTF\bin\adtf_datexporter.exe ^
  -import "F:\my_dat.dat" ^
  -stream RawCanFD -start 5000000 -end 6000000 ^
  -output "F:\my_dat_can.dat"
```

Mit

```
ADTF\bin\adtf_datexporter.exe ^  
-import "F:\my_dat.dat" ^  
-stream RawCanFD ^  
-ouput "F:\my_dat_can.dat"
```

Funktioniert der Export des gesamten CAN, inkl. Anzeige des Status in der Kommandozeile, doch möchte ich nur einen frei definierbaren Zeitbereich aus dem Gesamtdatenstrom. Der CAN liegt die ganze Zeit im Dat File an.

Viele Grüße
Steve

#10 - 2020-03-25 11:35 - hidden

Hallo Steve,

hab im Homeoffice leider keine 2.13.2 zur Hand, deshalb mit 2.14.2 getestet und dem example file darin:

```
c:\tools\adtf\2.14.2\bin>adtf_datexporter.exe -liststreams templates\example_test_file.dat
```

```
Streams in c:\tools\adtf\2.14.2\bin\templates\example_test_file.dat
```

```
Source: adtf.import.dat_file_source
```

```
  Name      Start  End
```

```
-----
```

```
video      113    14798822
```

```
  available sinks:
```

```
    BMP Video Sink: adtf.export.bmp
```

```
    CSV Sink: adtf.export.csv
```

```
NESTED_STRUCT  113    14805419
```

```
  available sinks:
```

```
    CSV Sink: adtf.export.csv
```

```
c:\tools\adtf\2.14.2\bin>adtf_datexporter.exe -import templates\example_test_file.dat -stream video new_video_  
dat_full.dat
```

```
..\..\..\..\source\src\services\datexporter\datexporter_srv.cpp(163): info 0x00000000: Processed 0%
```

```
..\..\..\..\source\src\services\datexporter\datexporter_srv.cpp(163): info 0x00000000: Processed 100%
```

```
c:\tools\adtf\2.14.2\bin>adtf_datexporter.exe -import templates\example_test_file.dat -stream video -start 500  
0000 -end 10000000 new_video_dat_5s_to_10s.dat
```

```
..\..\..\..\source\src\services\datexporter\datexporter_srv.cpp(163): info 0x00000000: Processed 34%
```

Abgesehen davon dass das Logging abschneidet klappts wunderbar und die beiden Files liegen im Application Dir, sind lauffähig und richtig beschnitten

Kannst du das mal mit dem example nachstellen ?

Und dann ggf. nochmal deinen call prüfen ?

PS: -ouput wäre a) ein Typo und b) kannst du dir das beim -import sparen und das .dat file direkt angeben.

#11 - 2020-03-25 13:45 - hidden

Hallo,

ich konnte das im ADTF 2.14.3 nachvollziehen und habe meinen Fehler gefunden.

Vielen Dank.

```
Name      Start  End
```

```
-----
```

```
RawLIN    -1    -1
```

```
  available sinks:
```

```
    none
```

```
RawCanFD  945454558  1944810016
```

Der Streamstart in meinem *.dat ist stets bei 945454558ys, demnach muss –start 945454558+5000000 –end 945454558+6000000 sein.

Das Ticket kann somit geschlossen werden.

Viele Grüße

Steve

#12 - 2020-03-26 12:21 - hidden

- *Subject changed from [ADTF 2.13.2] adtf_datexporter.exe to Notation of timestamps for the adtf_datexporter.exe*
- *Description updated*
- *Status changed from Customer Feedback Required to To Be Closed*
- *Resolution set to Solved Issue*
- *Platform Windows 7 64bit added*

#15 - 2020-07-07 16:33 - hidden

- *Project changed from 20 to Public Support*
- *Private changed from Yes to No*

#16 - 2020-07-07 16:41 - hidden

- *Status changed from To Be Closed to Closed*