

Public Support - Support Request #11808

EBPRODUCTSUPPORT-7228 Signal Provider missing functionality

2020-07-28 13:00 - hidden

Status:	Closed		
Priority:	Normal		
Category:			
Customer:	ELEKTROBIT	Product Issue Numbers:	https://www.cip.audi.de/jira/browse/ADEVTB-1871
Department:	SUPPORT	Affected Products:	ADTF Device Toolbox 3.2.0
Requester's Priority:	Normal	Platform:	Windows 10 64bit
Support Level:	3rd Level	Topic:	DeviceTB::CAN
Resolution:	Product Issue Opened	FAQ Links:	

Description

Hallo Supporterkollegen,

Ein Kollege war mal wieder fleissig und hat folgende Probleme gefunden (konnte auch reproduziert werden):

I. Der CAN Signal Provider schaut sich nur Sample Streams an, die mit dem ADTF DAT File Player verbunden sind.

Zum Nachstellen:

Das Device Toolbox Demo-Projekt öffnen
In der Session Demo CAN Config Encoder Session:

- Im Streaming Graph einen neuen Sample Stream mit output_can verbinden
- Im System Editor einen CAN Signal Provider und einen Signal Tree View hinzufügen

Nach dem Launch stellt man fest, dass im Signal Tree View nur am neuen Sample-Stream Signale vorhanden sind.

II. Beim Replay von ADTF 2 CAN Streams fehlt die Mediadescription. (CAN Stream in .dat file).

Zum Nachstellen:

Einen ADTF Dat File-Player zum Abspielen eines .dat files mit CAN bringen. Den CAN-Pin mit dem Demo-Qt-Media-Description-Display verbinden.

Das ging schonmal in der Vorversion, daher ein neuer Bug. Oder soll das so sein?

III. In der Vorversion (ADTF 3.6 + Device TB 3.1) gab es die Funktionalität das der CAN-Channel beim Abspielen von ADTF 2 CAN streams in den Samples von 0 auf 1 geändert wurde. Das funktioniert bei mir nicht mehr.

Das Feature ist nicht mehr in der SDK doc dokumentiert sondern im Guide
file:///C:/ADTF/orig/adtf-3.7.1/addons/devicetoolbox/doc/guides/whats_new.html
Weil Punkt II nicht mehr geht muss man sich zur Überprüfung einen Filter schreiben, der den Channel ausgibt.
Auch wenn ich Debug starte bekomme ich nicht angezeigt, dass das `adtf_devtb_2_deserializer.adtffileplugin` geladen wird. Muss das in einem bestimmten Direktory stehen? (Es gibt keine Meldung von einem `.adtffileplugin`)

Lg Flo

History

- #1 - 2020-07-28 14:09 - hidden
- Status changed from New to In Progress
 - Topic set to DeviceTB::CAN
 - Support Level changed from 2nd Level to 3rd Level
 - Customer set to ELEKTROBIT
 - Department set to SUPPORT

- Affected Products ADTF Device Toolbox 3.2.0 added

#2 - 2020-07-28 16:01 - hidden

- Status changed from In Progress to Customer Feedback Required

Hallo,

vielen Dank schon einmal für dein Feedback.

Zu I.

Versuchst du einen Ausgangspunkt mit zwei SampleStreams zu verbinden? Das ist nicht möglich

Zu II.

Wenn es vorher schon einmal ging, dann soll das meiner Ansicht nach weiterhin funktionieren. Soll mittels des Qt-Mediadescriptions-Displays die CAN-DDL angezeigt werden?

Zu III.

Ist das adtffileplugin automatisch beim hinzufügen des can-service mitgeladen? Die Doku zum what's new ist in die Guides gewandert.

https://support.digitalwerk.net/adtf_addons/adtf-device-toolbox/v3/devicetoolbox_html/page_devtb2_support.html

https://support.digitalwerk.net/adtf_addons/adtf-device-toolbox/v3/guides/whats_new.html

#3 - 2020-07-29 16:41 - hidden

- File 11808-devtb2Support.zip added

Ich habe einmal eine Session zusammengefasst und auch ein kurzes Video hinzugefügt

#4 - 2020-08-07 09:49 - hidden

Hi zusammen,

schonmal vielen Dank für die Antwort. Leider ist der Kollege, dem das Problem aufgefallen ist, derzeit im Urlaub, also bitte noch nicht schließen, ihr bekommt Feedback sobald der Kollege zurück ist.

#6 - 2020-08-25 09:10 - hidden

Feedback auf Ende August verlängert

#7 - 2020-08-25 14:50 - hidden

- File DeviceToolbox3.3.PNG added

Hallo Florian,

Zu Punkt I. Hier war der Bugreport, dass sich der CAN Signal Provider offenbar nicht alle Sample Streams anschaut, von denen er Signals Providen sollte.

Das ist auch noch bei der Device Toolbox 3.3 so. Hier stellt sich das offenbar so dar: Sample Streams im Streaming Graph werden Provided, Sample Streams im Filter Graph nicht.

Siehe Pic: Die CAN Config Decoder Session wurde so ergänzt, dass auch ein Demo CAN Coder Generator 2 im Filtergraph vorkommt. (neben dem Original im Streaming Graph)

Der CAN Trace View liefert traces von allen CAN Sample Streams im Filtergraphen, aber nicht im Streaminggraphen. Der Signal Tree View findet aber nur das Signal vom Sample Stream im Streaming Graph.

Das ist natürlich etwas blöd für ADTF 3.8, wo es im Normalfall keine Streaminggraphen gibt. Wie ist da eigentlich der Release Status? DTB 3.3 fehlt noch auf eurer Homepage, ist aber per Email angekündigt worden. Wurde der Release zurückgezogen?

Die anderen Themen mit den CAN Streams aus .dat Files muss ich mir nochmal genauer anschauen, brauche da noch etwas Zeit, Florian konnte es allerdings auch nachvollziehen.

#8 - 2020-09-03 09:19 - hidden

- File Launcher_Crash1.PNG added

- File Launcher_Crash2.PNG added

Hallo allerseits,

hab nun angefangen mich wieder mit dem weiteren Thema zu beschäftigen: .dat file mit CAN stream.

Zunächst mal scheint ein weiterer Bug zu existieren. Wenn ich Benedicts Example-Projekt launche, crashed der Launcher beim Switch auf Runlevel 2 in der Release Version. (Popups attached). Das zweite Stacktrace-Popup erhalte ich allerdings nur beim Launch mit der GUI-Control (nicht mit mit

Control oder reiner Launcher). In der Debug-Version startet das Projekt, allerdings sieht das Fenster des CAN trace view im rl 2 rein grau aus. Ich tippe deshalb auf ein Problem beim CAN-Trace-View.

In der Debug-Version läuft das Projekt und Media-Description Display und CAN Channel ist 1 angezeigt. DBC-Verwendung im CAN Trace View funktionieren. Also hier alles gut, wie es sein sollte.

Ich bau das jetzt nochmal nach.

Grüße,
Jochen

#9 - 2020-09-03 12:40 - hidden

- *File redo4th.7z added*

Hallo nochmal,

nun habe ich das ganze nachgebaut.

Mit dem example_test_file_can.dat aus eurem Projekt klappt es (ohne crash). Der DemoQtMediaDescription Display zeigt die CAN Samples an, der CAN Channel wird auf 1 geändert.

Aber es scheint auch noch abhängig vom verwendeten .dat file. Verwendet man das rawcan-example aus ADTF 2.9.1 (example_test_file.dat), also der alte can Datentyp, wird keine Description angezeigt.

Nun habe ich anbei noch ein Projekt mit einem .dat file schon mit neuem CAN Daten-Typ, wo auch keine Media-Description angezeigt wird. Woran kann das liegen? Der CAN stream in ADTF 2 wird voll unterstützt (bezüglich CAN Signal, CAN Config Codec Ext). Einziger Unterschied, den ich in ADTF 2 sehen konnte, ist, dass im Filter Trace View beim Breakpoint hier auch keine Media-Description angezeigt wird, im Gegensatz zum example_test_file.dat (obwohl auch dort kein .description beiliegt). Vermutlich wird der CAN Stream in ADTF 3 nicht richtig erkannt und dementsprechend der Channel nicht geändert, worauf dann CAN Trace View und CAN Signal Provider nicht funktionieren.

Zu Punkt III: wie erkennt man, ob das ADTF File Plugin geladen wird?

Grüße,
Jochen

#10 - 2020-09-07 10:40 - hidden

- *File dattool-Ausgabe.png added*

Ich habe mir das angehängte Projekt und das zugehörige Dat-File einmal angeschaut.

```
FreeCommander - DOS
C:\Users\bh\Work\Support\11808-devtb2Support>C:\.conan\1dad2\1\pkg\adtf_file\bin\adtf_dattoold.exe --plugin C:\devtb\ma
ster\package\bin\debug\adtf_devtb_2_deserialized.adtffileplugin --liststreams redo4th\adtfsessions\example.dat
adtfdat:
  front_cam:
    type: adtf/image
    processors:
    time range (ns): [3184000, 9961023000]
    items: 205
  can:
    type: adtf2/legacy
    processors:
    time range (ns): [0, 9995239000]
    items: 3078
  nmea:
    type: adtf/plaintype
    processors:
    time range (ns): [24547000, 9081361000]
    items: 60
C:\Users\bh\Work\Support\11808-devtb2Support>C:\.conan\1dad2\1\pkg\adtf_file\bin\adtf_dattoold.exe --plugin C:\devtb\ma
ster\package\bin\debug\adtf_devtb_2_deserialized.adtffileplugin --liststreams C:\Users\bh\Work\DevTB\2_7_1\vw\src\examp
les\src\demo_IEEE_cangenerator\configuration\files\IEEEFloatDouble10perSecond.dat
adtfdat:
  output:
    type: adtf/devicetb/can
    processors:
    time range (ns): [159962000, 15259955000]
    items: 152
C:\Users\bh\Work\Support\11808-devtb2Support>
```

Hier wird dem can-Sample-Stream bereits der Typ adtf2/legacy gegeben, das kommt daher, dass der Datenstrom mit einer normaler cMediaType-Instanz serialisiert wurde und nicht mit einer cMediaTypeCan-Instanz. Auch die Samples sind normale cMediaSamples-Instanzen und nicht cMediaSampleCAN. Daher wird der Datenstrom von den adtffiledeserializern nicht als CAN-Datenstrom erkannt. Mit welchen Komponenten wurde dieser Datenstrom erzeugt?

Workaround ist leider nur ein eigener Filter, der den Datentyp konvertiert. Hier kann auch dann gleich der Channel mit angepasst werden.

#12 - 2020-09-07 19:03 - hidden

Hi Benedict,

vielen Dank für die Aufklärung bezüglich des .dat Files! Der Punkt mit dem ADTF 2 DAT Support ist damit geklärt.

Kannst Du Dir Bitte noch die Bug-Reports anschauen, die ich weiter oben kommentiert hatte?

Der Bug-Report zu Punkt III ist mir am Wichtigsten (Posting vor 13 Tagen, 3 weiter oben). Das hat sich bei ADTF 3.8 leider nochmal verschlechtert.

Dann noch der Bug, dass mein Launcher beim Abspielen Deines Projekts crashed (Posting for 4 Tagen, 2 weiter oben).

Grüße,
Jochen

#13 - 2020-09-07 19:07 - hidden

P.S. ich meine den Bug zu Punkt I nicht III, sorry für den Typo.

#14 - 2020-09-09 12:24 - hidden

Es ist schon korrekt, dass die SampleStreams in der Devicetoolbox 3.x nur die Samples aus dem Streaming Graph aufnehmen. Der Hauptanwendungsfall lag hierbei im Aufnahme des Bus-Signals im Streaming Graph und Verarbeitung (Decodierung) im Filtergraph

#15 - 2020-09-10 13:11 - hidden

Hallo Benedict,

Wie gesagt so durchgängig ist das nicht. Der Signal Provider provided nur Signals vom CAN Streams im Streaming graph. Der Can Trace View nur die im Filter Graph (Device Toolbox 3.3). Das macht in ADTF 3.8 erst mal gar keinen Sinn.

In der Device Toolbox 2.2 war es ja nochmal anders (vergleiche mein erstes Beispiel, der Ursprüngliche Bug report: Klar geht Parallel-Schaltung von

Sample Streams am Pin nicht, du musst sie hintereinander schalten. Vom zweiten werden keine Signale provided).

Das sollte IMHO so wie in ADTF 2 funktionieren, dass von beiden Services alle Streams/Pins mit CAN streams unterstützt werden.

Machst Du da bitte ein Bug/Improvement Ticket auf?

Und für den anderen Bug?

Grüße,
Jochen

#16 - 2020-09-21 16:16 - hidden

Hallo,

gibt es Neuigkeiten?

Grüße,
Jochen

#17 - 2020-09-30 11:31 - hidden

- Status changed from Customer Feedback Required to To Be Closed
- Resolution set to Product Issue Opened
- Product Issue Numbers set to <https://www.cip.audi.de/jira/browse/ADEVTB-1871>
- Platform Windows 10 64bit added

Entschuldigung, dass es so lange gedauert hat.

Aktuell erfasst der CAN Signal Provider lediglich Outputpins, von IDataBinding-Objekten im Streaminggraph, die auch mit einem Sample-Stream verbunden sind.

Das Verhalten sollte allerdings geändert werden. ich habe dazu folgendes Change-Ticket erstellt.

<https://www.cip.audi.de/jira/browse/ADEVTB-1871>

#18 - 2020-09-30 12:36 - hidden

Hi Benedict,

danke Dir! Auch für die Erklärung. Damit ist klar, dass es nur für die Sample Streams funktioniert, die direkt mit dem Output Pin einer Streamingsource im Streaming Graph verbunden sind.

Der 2. Bug. Dem Crash mit dem Can Trace View beim Abspielen Deines Projekts. Magst Du dem nachgehen?

Deutet ja auf einen nicht abgefangen Initialisierungsfehler hin.

Grüße,
Jochen

#19 - 2020-09-30 13:44 - hidden

- File Release-Files.gif added

Hat etwas gedauert, bis ich den Fehler nachvollziehen konnte.

Das QWidget von ADTF (Release) versucht das CAN-Trace View von der Device-Toolbox zu öffnen. In dem Projekt, dass ich geschickt habe, war aber nur eine Debug-Variante der Device Toolbox. Aus dem Grund hat er unter Plugins(Release) auch die Debug-Variante eingetragen. Bin "eigenen" Filtern funktioniert das auch, aber das ist eine Qt-Applikation.

Siehe dafür das angehängte Video

#20 - 2020-10-06 16:54 - hidden

Hi Benedict,

tatsächlich das wars.

Ich glaube denselben Fehler werden wir noch öfter sehen. Passiert offenbar regelmäßig, wenn man ein Qt-Filter/Service nur debug kompiliert.

Evtl. wäre eine Warning oder Information fürs Validation Protokoll angebracht.

Sonst könnte man von meiner Seite das Ticket schließen.

Grüße,

Jochen

#21 - 2020-10-07 12:23 - hidden

Hallo Jochen,

Evtl. wäre eine Warning oder Information fürs Validation Protokoll angebracht.

Leider ist das ein explizites Problem und diese können wir nicht alle behandeln.
Es ist auch kein ADTF oder Qt Problem, sondern grundsätzlich ein Binary Build Type Issue (das Netz ist voll davon), d.h. hier sprechen wir von ADTF und QT, ein ander mal von weiteren Build Abhängigkeiten.
Wir können hier eigentlich nur die Entwickler und Anwender sensibilisieren, deshalb steht es ganz oben in der Doku:

- https://support.digitalwerk.net/adtf/v3/adtf_html/page_system_requirements.html

Was wir zu ADTF 3.9 gemacht haben, wir loggen den Build Type, wenn ein adtfplugin geladen wird, ebenso kann es mittels adtf control (listcomponents oder createreport) abgefragt werden.
Dann hat man zumindest einen Ansatz bei der Fehlersuche.

In die Fehlermeldung von Qt können wir uns leider nicht reinhacken, ebenso macht es wie oben geschrieben leider keinen Sinn, im Validation Protocol zu warnen, da es durchaus ein Anwendungsfall ist, Plugins zu mischen (weil zb vom Dienstleister nur Release kommt), der CE aber beides designen muss. Das ist im System Editor ja ersichtlich, was er nimmt. Deshalb wäre es am Ende für den einen eine Warnung, für den nächsten ein Fehler und für mich zum bsp. Überflüssig...

#23 - 2020-10-09 12:51 - hidden

- Subject changed from EBPRODUCTSUPPORT-7228 ADTF Device Toolbox 3.2 - CAN Signal Provider missing functionality to EBPRODUCTSUPPORT-7228 Signal Provider missing functionality
- Private changed from Yes to No

#24 - 2022-01-17 08:17 - hidden

- Status changed from To Be Closed to Closed

Files

11808-devtb2Support.zip	20.3 MB	2020-07-29	hidden
DeviceToolbox3.3.PNG	93.2 KB	2020-08-25	hidden
Launcher_Crash1.PNG	10.5 KB	2020-09-03	hidden
Launcher_Crash2.PNG	37.2 KB	2020-09-03	hidden
redo4th.7z	7.65 MB	2020-09-03	hidden
dattool-Ausgabe.png	29.9 KB	2020-09-07	hidden
Release-Files.gif	48.3 MB	2020-09-30	hidden