

Public Support - Support Request #10989

EBPRODUCTSUPPORT-6631 Sending CAN data using Support TB fails

2020-04-07 09:00 - hidden

Status: Closed	
Priority: Normal	
Category:	
Customer: ELEKTROBIT	Product Issue Numbers: https://www.cip.audi.de/jira/browse/ADEVTB-1796
Department: SUPPORT	Affected Products: ADTF 2.14.3, ADTF 3 Support Toolbox 3.1.0, ADTF 3.6.2, ADTF Device Toolbox 2.6.2, ADTF Device Toolbox 3.1.0
Requester's Priority: Normal	Platform: Ubuntu 16.04 64bit, Windows 10 64bit
Support Level: 3rd Level	Topic: DeviceTB::CAN
Resolution: Product Issue Opened	FAQ Links:
Description	
Supportanfrage	
Mein Kollege nutzt folgende Versionen:	
ADTF 3.6.2 ADTF Device Toolbox 3.1	
ADTF 2.14.3 ADTF 3 support toolbox 3.1 ADTF Device Toolbox 2.6.2	
Er benutzt die ADTF 3 support toolbox um einen CAN stream von ADTF 2 nach 3 zu schicken mit der ADTF_3_Host_Only_Pipe_Sink. In ADTF 3 gibt es Probleme wenn er mit IPC Host only Source empfangen will. Es gibt dann den Fehler: "no sample deserializer factory for adtf.sample.can_message.adtf2_support.serialization.adtf.cid available".	
In ADTF 3 fügte er dann den DevTB2 Support Service hinzu was die Fehlermeldung veränderte: 20 warning 01.04.2020 09:56 error occurred in 'default_streaming_graph.IPC Host Only Receiver' (category: 'ipc_error'): Result code '-36 '(ERR_UNKNOWN_FORMAT) - Received unsupported message: 1948281961	
"The error occurs already after initialization of the ADTF 2 Launcher (running not needed). Adding the CAN Support Service does not change it.	
IMHO we get some information from the original error 20 warning 2020-04-03 08:59:21 error occurred in 'default_streaming_graph.IPC Host Only Receiver' (category: 'ipc_error'): Result code '-38 '(ERR_FAILED) - no sample deserializer factory for 'adtf.sample.can_message.adtf2_support.serialization.adtf.cid' available Unable to read from input stream. File: s:\cs\ce94ae\1\source\src\libraries\ipcsdk\src\ipc_sample_source.cpp<File://%20s:csce94ae1sourcesrclibrariesipcsdksrcipc_sample_source.cpp> [Line: 173] [Func: adtf::ipc::ant::detail::cIPCSampleSource::ReadStreams] File: s:\cs\ce94ae\1\source\src\libraries\ipcsdk\include\adtfipcsdk\ipc_sample_source.h<File://%20s:csce94ae1sourcesrclibrariesipcsdkincludeadtfipcsdkipc_sample_source.h> [Line: 202] [Func: adtf::ipc::ant::ipc_sample_source<class cPipeInputStream>::Read] error_handling.cpp(62)	
It looks like that the ADTF 3 Support Library provides the Stream Type ID 'adtf.sample.can_message.adtf2_support.serialization.adtf.cid'. Maybe would just be better if the suffix could be removed. Anyway I do not understand why the IPC Host Only Source cares about the stream type. Just ignoring it and writing an adapter filter would at least work.	
Is there a workaround? Should this work?	
Then one more question:	
In ADTF 2 often the CAN channel 0 was used. The CAN Support Service in the ADTF 3.1 Device Toolbox currently supports CAN	

Channels 1 to 9. I saw in some planned ticket that there will be up to 16 CAN channels supported. Will the CAN channel 0 also be supported?"

Lösung

Wird mit Device TB 3.2.0 gefixt (ADEVTB-1796)

History

#1 - 2020-04-07 11:48 - hidden

- Project changed from Public Support to 7

- Status changed from New to In Progress

- Topic set to ADTF3SupportTB::Common

- Customer set to ELEKTROBIT

- Department set to SUPPORT

- Affected Products ADTF 2.14.3, ADTF 3 Support Toolbox 3.1.0, ADTF 3.6.2, ADTF Device Toolbox 2.6.2, ADTF Device Toolbox 3.1.0 added

#2 - 2020-04-09 13:39 - hidden

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

#4 - 2020-04-15 17:14 - hidden

Thema muss noch weiter untersucht werden, Verhalten ist reproduzierbar

#5 - 2020-04-23 08:21 - hidden

Ich denke das liegt am gleichen Problem wie ADEVTB-1740 nur für CAN statt CANFD. Der Legacy Deserialisierer ist leider einfach falsch implementiert und liest zuviel/falsch aus dem Bytestream und dann kommts zu der Fehlermeldung.

Als Möglicher Workaround, fällt mir nur ein, auf ADTF2 Seite mit dem "CAN Legacy Filter Out" Filter die Daten vor dem Versenden noch in das "uralte" CAN Format zu konvertieren, da könnten die Deserialisierer funktionieren.

@Flo kannst Du das in deinem Setup ausprobieren?

@Bene, erstellst Du da bitte das entsprechende Ticket in der Device Toolbox und checkst auch gleich noch alle anderen Deserialisierer auf das gleiche Problem?

Grüße,

Martin

#7 - 2020-04-23 09:34 - hidden

In ADTF 2 often the CAN channel 0 was used. The CAN Support Service in the ADTF 3.1 Device Toolbox currently supports CAN Channels 1 to 9. Will the CAN channel 0 also be supported?"

- In der Device Toolbox 3.x wird der Channel 0 nicht mehr "angeboten". Die neue Zählweise beginnt bei 1. Das ist der Anforderung einiger Kunden geschuldet, die eine Analogie zum Vector Hardware Config Tool gewünscht haben. (Gibt dort sicher zwei verschiedene Meinungen dazu)
- Die Deserialisierer von CAN und CAN FD werden aber den Channel erhöhen und dementsprechend deserialisieren

I saw in some planned ticket that there will be up to 16 CAN channels supported

- Das ist korrekt. Die Anzahl wird in bald erhöht. Ist das ein valider Anwendungsfall für Sie?

#9 - 2020-04-23 11:20 - hidden

Kleine Anmerkung/Nachtrag zu [#10989#note-7](#):

Aufzeichnungen aus ADTF 2.x mit Channel 0 werden in ADTF 3.x automatisch auf Channel 1 gemapped und sind verwendbar. Steht auch so in der Doku und in der Logausgabe im genannten Fall.

#12 - 2020-04-27 18:36 - hidden

Als Möglicher Workaround, fällt mir nur ein, auf ADTF2 Seite mit dem "CAN Legacy Filter Out" Filter die Daten vor dem Versenden noch in das "uralte" CAN Format zu konvertieren, da könnten die Deserialisierer funktionieren.

@Flo kannst Du das in deinem Setup ausprobieren?

@Martin: Dann bekomme ich zwar keine Fehlermeldungen mehr aber es kommen auch keine Samples an... es passiert einfach nichts

#14 - 2020-04-28 08:43 - hidden

- *Product Issue Numbers set to <https://www.cip.audi.de/jira/browse/ADEVTB-1796>*

#19 - 2020-05-06 09:07 - hidden

- *Resolution set to Product Issue Opened*
- *Status changed from In Progress to To Be Closed*
- *Platform Ubuntu 16.04 64bit, Windows 10 64bit added*

Das ist doch mit bereits gefixed worden, oder?

#20 - 2020-05-06 10:25 - hidden

- *Project changed from 7 to Public Support*
- *Subject changed from EBPRODUCTSUPPORT-6631 [EB internal] ADTF 3 Support Toolbox - CAN Sample to EBPRODUCTSUPPORT-6631 Sending CAN data using Support TB fails*
- *Description updated*
- *Private changed from Yes to No*
- *Topic changed from ADTF3SupportTB::Common to DeviceTB::CAN*

Wird mit Device TB 3.2.0 gefixt (ADEVTB-1796)

#21 - 2020-05-07 12:32 - hidden

Hallo Florian,

durch die ganzen Coronamaßnahmen hänge ich etwas hinterher. Vielleicht könnten die Punkte in der Dokumentation noch präserter vermerkt werden:

- Auch den Service dokumentieren, da wir den Punkt nur unter Whats new gefunden haben
 - Könnt ihr den richtigen Bezeichner für den Service nehmen?
 - Außerdem könntn ihr folgendes in der Doku prüfen: "The "devtb_2_deserializer" plugin is loaded automatically by the appropriate bus service"? Wir mussten den Service manuell in das Projekt einbinden, stimmt die Dok u an der Stelle?

Und hier gibt es gleich auch den Anwendungsfall für die 16 CAN Channels, da ADTF 2 CAN Channels 0 bis 9 unterstützt und ADTF 3 nur 1 bis 9 ist aktuell unklar, wie der 10. ADTF 2 CAN Channel gemappt wird. Das kann man mit 10 ADTF 3 CAN Channels lösen.

Danke und Grüße,
Anja

#22 - 2020-05-07 12:34 - hidden

- *Status changed from To Be Closed to In Progress*

#23 - 2020-05-07 16:38 - hidden

Vielen Dank für den Hinweis. Ich gehe die einzelnen Punkte einmal durch

#24 - 2020-05-15 14:24 - hidden

- *File media_type_sample_class_ids.xml added*

Hinweis: Wir haben es nun für alle Datentypen mit aktuellen master noch einmal getestet und es geht nun, wird also mit Device TB 3.2.0 funktionieren.
Wenn ihr EXT oder RAW Typen statt CAN Data übertragen wollt, müsst ihr allerdings das Mapping File in der Support TB patchen, ich leg das File

auch nochmal zum Downloader und wir werden das bei Gelegenheit in einer Patch Version anpassen... hier hat sich leider ein Fehler eingeschlichen

#25 - 2020-05-29 17:33 - hidden

- Status changed from In Progress to To Be Closed

#26 - 2020-07-07 12:49 - hidden

- Status changed from To Be Closed to Closed

Files

20200403090540438_image002.jpg	363 KB	2020-04-07	hidden
media_type_sample_class_ids.xml	756 Bytes	2020-05-15	hidden