Public Support - Support Request #11154

How to read chunks of type CAN using ADTF File Library

2020-05-05 12:15 - hidden

Status:	Closed		
Priority:	Normal		
Category:			
Customer:	DAIMLER	**Product Issue Numbers:**	
Department:	RDI/TBP	**Affected Products:**	ADTF Device Toolbox 3.1.0, ADTF File Library 0.6.1 (BETA)
Requester's Priority:	Normal	**Platform:**	Other Linux 64bit, Windows 10 64bit
Support Level:	2nd Level	**Topic:**	FileLibrary::Common
Resolution:	Solved Issue	**FAQ Links:**	

Description

Supportanfrage

We are following CANWriter example to understand the CAN structure and similarly we are trying to read CAN block into the CAN structure using readNextChunk API where stream ID is assigned to CAN stream ID and also tried to read CAN structure data using readChunk API.

Am I using the correct API to extract information or please suggest which APIs should I use to read CAN block into CAN structure.

Lösung

You cannot use asn IndexedFileReader to access the sample data, as chunks contain only serialized versions of the samples. You need to use and adtf_file::Reader as is shown in the "File Access" example (https://support.digitalwerk.net/adtf_libraries/adtf-file-library/v0/html/a02199.html). This will give you the deserialized sample data. It is important, that you load the adtf_devtb_2_deserializerd.adtffileplugin if you want to access ADTF2 Dat files with CAN streams. The File Access Examples does this when specifying the plugin on the command line.

Then you can access the sample data as is done in the "access_file_data" function in the example.

History

#1 - 2020-05-05 13:51 - hidden

- Project changed from Public Support to 9

- Status changed from New to Customer Feedback Required

- Topic set to DeviceTB::CAN

Hello Vidhi Patel,

are you using ADTF 2.x or ADTF 3.x?
Please also tell us which version of the ADTF Device Toolbox is used?
To which "CANWriter example" did you refer?

Thank you

#2 - 2020-05-06 07:45 - hidden

Hello Matthias,

We are using ADTF 2.x
ADTF device toolbox: 3.1.0
CANWriter example: https://support.digitalwerk.net/adtf_libraries/adtf-file-library/v0/html/a02196.html

Thanks,
Vidhi

#3 - 2020-05-06 08:00 - hidden

- Status changed from Customer Feedback Required to In Progress

- Topic changed from DeviceTB::CAN to FileLibrary::Common

- Affected Products ADTF File Library 0.6.1 (BETA) added

#4 - 2020-05-06 08:14 - hidden

- Affected Products ADTF Device Toolbox 3.1.0 added

#5 - 2020-05-06 11:00 - hidden

- File code snippet.png added

Hello,

I have included this file ‘can_stream_type_definitions.h’ in ADTF_fileaccess.cpp example code and tried to used tCANData structure shown as attached screenshot where last argument ‘1’ is CAN stream ID. And trying to access tCANData structure variables. (example: sData.ui32Id)

#8 - 2020-06-03 09:36 - hidden

Hi vidhi patel,

you cannot use asn IndexedFileReader to access the sample data, as chunks contain only serialized versions of the samples. You need to use and adtf_file::Reader as is shown in the "File Access" example (https://support.digitalwerk.net/adtf_libraries/adtf-file-library/v0/html/a02199.html). This will give you the deserialized sample data. It is important, that you load the adtf_devtb_2_deserializerd.adtffileplugin if you want to access ADTF2 Dat files with CAN streams. The File Access Examples does this when specifying the plugin on the command line.

Then you can access the sample data as is done in the "access_file_data" function in the example.

Please get back to us if you need further assistance.

Regards,

Martin

#9 - 2020-06-09 09:43 - hidden

- Status changed from In Progress to To Be Closed

- Resolution set to Solved Issue

- Platform Other Linux 64bit, Windows 10 64bit added

#10 - 2020-06-09 12:12 - hidden

- Subject changed from Reading chunk (CAN block) APIs query to How to read chunks of type CAN using ADTF File Library

- Description updated

#11 - 2020-07-07 16:18 - hidden

- Project changed from 9 to Public Support

- Private changed from Yes to No

#12 - 2020-07-07 16:41 - hidden

- Status changed from To Be Closed to Closed

Files

code snippet.png	35.8 KB	2020-05-06	hidden