

Public Support - Support Request #19542

Using Dynamic vectors as inputs

2023-03-01 13:45 - hidden

Status:	Closed	Product Issue Numbers: Affected Products: ADTF 3.14.3, ADTF 3.15.3 Platform: Windows 10 64bit Topic: ADTF::FilterSDK FAQ Links:
Priority:	Normal	
Category:		
Customer:	CARIAD	
Department:	T3-HD	
Requester's Priority:	Normal	
Support Level:	2nd Level	
Resolution:	Solved Issue	

Description

Support Request:

I want to understand if i can send dynamic vectors as inputs to my ADTF filters as Streams. I have this question because I want to know then how do I configure the input pin in this case? I also want to keep the input type anonymous. Additionally, how would my description file look in this case?

Please let me know.

Solution:

I want to understand if i can send dynamic vectors as inputs to my ADTF filters as Streams
I also want to keep the input type anonymous.

Sure, even though that implies you will then also have to work with the raw sample buffers, and you can't use any of the strongly typed (and size checking) helpers like `sample_data<T>` and alike.

I have this question because I want to know then how do I configure the input pin in this case?

Check `/src/examples/src/adtf/filters/standard_filters/custom_stream_type/stream_type_custom_sensor.h` in your ADTF copy for an example, and https://support.digitalwerk.net/adtf/v3/adtf_html/classadtf_1_1streaming_1_1ant_1_1stream_meta_type.html for the documentation of the template your type definition needs to work with.

Additionally, how would my description file look in this case?

You don't have any. Unstructured data with arbitrary dynamic elements can't be described in DDL4. At most it will be able to describe a preamble to your dynamic data which has a static layout (think e.g. Ethernet headers), but everything starting from the first dynamic byte can't be expressed. This will also render it completely unusable with any type of the display, they will not give you any information other than "there was a sample".

Can ADTF in general have such a dynamic input?

Yes, that is supported. Raw buffer access (without using the higher level `sample_data` helpers etc.) is possible:

```
//Buffer access is permitted as long as the (read) lock is held
ucom::object_ptr_shared_locked<const streaming::ISampleBuffer> pBuffer;
RETURN_IF_FAILED(pSample->Lock(pBuffer));
your_call(/* raw, unaligned byte-buffer */ static_cast<const uint8_t*>(pBuffer->GetPtr()),
/* size of buffer in bytes */ pBuffer->GetSize());
```

You will need to deal with reinterpret-casts or safe unaligned memory accesses yourself, but I expect you are versed in how to operate with C-style raw buffers.

History

#1 - 2023-03-01 14:07 - hidden

- Project changed from Public Support to 30
- Status changed from New to In Progress
- Customer set to CARIAD
- Department set to T3-HD
- Topic set to ADTF::FilterSDK

Hello Divyapriya,

in advance can you please tell us the ADTF 3.x version in use.
Also, can you please give use a few more details and the use case?

Regards
Matthias

#2 - 2023-03-01 16:15 - hidden

Hello Divyapriya,

I want to understand if i can send dynamic vectors as inputs to my ADTF filters as Streams
I also want to keep the input type anonymous.

Sure, even though that implies you will then also have to work with the raw sample buffers, and you can't use any of the strongly typed (and size checking) helpers like `sample_data<T>` and alike.

I have this question because I want to know then how do I configure the input pin in this case?

Check `/src/examples/src/adtf/filters/standard_filters/custom_stream_type/stream_type_custom_sensor.h` in your ADTF copy for an example, and https://support.digitalwerk.net/adtf/v3/adtf_html/classadtf_1_1streaming_1_1ant_1_1stream_meta_type.html for the documentation of the template your type definition needs to work with.

Additionally, how would my description file look in this case?

You don't have any. Unstructured data with arbitrary dynamic elements can't be described in DDL4. At most it will be able to describe a preamble to your dynamic data which has a static layout (think e.g. Ethernet headers), but everything starting from the first dynamic byte can't be expressed. This will also render it completely unusable with any type of the display, they will not give you any information other than "there was a sample".

Best regards,
Andreas

#3 - 2023-03-01 16:36 - hidden

- Status changed from In Progress to Customer Feedback Required

#4 - 2023-03-02 08:00 - hidden

Hi,

I am using ADTF3.14.3. The use case is that my input is a dynamic vector so I do not know the exact size of my input struct, I want it to be dynamic.
Can ADTF in general have such a dynamic input?

Thank you.

#5 - 2023-03-02 08:28 - hidden

- File deleted (image001.png)

#6 - 2023-03-02 08:28 - hidden

- Description updated

#7 - 2023-03-02 08:28 - hidden

- File deleted (image002.png)

#8 - 2023-03-02 08:29 - hidden

- File deleted (image003.png)

#9 - 2023-03-02 08:29 - hidden

- File deleted (image004.png)

#10 - 2023-03-02 08:29 - hidden

- File deleted (image005.png)

#11 - 2023-03-02 08:29 - hidden

- File deleted (image006.png)

#12 - 2023-03-02 08:29 - hidden

- File deleted (image007.png)

#13 - 2023-03-02 08:29 - hidden

- File deleted (image008.png)

#14 - 2023-03-02 08:29 - hidden

- File deleted (image009.png)

#15 - 2023-03-02 08:29 - hidden

- File deleted (image010.png)

#16 - 2023-03-02 08:29 - hidden

- File deleted (image011.png)

#17 - 2023-03-02 08:29 - hidden

- File deleted (image012.png)

#18 - 2023-03-02 08:29 - hidden

- File deleted (image013.png)

#19 - 2023-03-02 08:30 - hidden

- Status changed from Customer Feedback Required to In Progress

#21 - 2023-03-03 09:51 - hidden

- Status changed from In Progress to Customer Feedback Required

Hello Divyapriya,

Can ADTF in general have such a dynamic input?

Yes, that is supported. Raw buffer access (without using the higher level sample_data helpers etc.) is possible:

```
//Buffer access is permitted as long as the (read) lock is held
ucom::object_ptr_shared_locked<const streaming::ISampleBuffer> pBuffer;
RETURN_IF_FAILED(pSample->Lock(pBuffer));
your_call(/* raw, unaligned byte-buffer */ static_cast<const uint8_t*>(pBuffer->GetPtr()),
/* size of buffer in bytes */ pBuffer->GetSize());
```

You will need to deal with reinterpret-casts or safe unaligned memory accesses yourself, but I expect you are versed in how to operate with C-style raw buffers.

Best regards,
Andreas

#26 - 2023-03-06 10:48 - hidden

Hello Divyapriya,

please give us feedback until tomorrow.

Regards

Matthias

#27 - 2023-03-06 10:52 - hidden

Hi Support team,

I want to understand if i can send dynamic vectors as inputs to my ADTF filters as Streams. I have this question because I want to know then how do I configure the input pin in this case? I also want to keep the input type anonymous. Additionally, how would my description file look in this case? Please let me know.

#28 - 2023-03-06 11:06 - hidden

Hello Divyapriya,

have you read the information from Andreas?

He already answered you about this topic.

Please see

[#note-2](#) and [#note-21](#) for details

There you can find info for examples within ADTF and about Raw buffer access.

Regards

Matthias

#29 - 2023-03-08 08:30 - hidden

hidden wrote in [#note-28](#):

Hello Divyapriya,

have you read the information from Andreas?

He already answered you about this topic.

Please see

[#note-2](#) and [#note-21](#) for details

There you can find info for examples within ADTF and about Raw buffer access.

Regards

Matthias

Hi,

My email replies do not reach you some how. Please close this ticket. Thank you for the information.

BR,

Divya

#30 - 2023-03-08 09:24 - hidden

- *Description updated*

- *Status changed from Customer Feedback Required to To Be Closed*

- *Resolution set to Solved Issue*

- *Affected Products ADTF 3.14.3, ADTF 3.15.3 added*

- *Platform Windows 10 64bit added*

Thank you for the Feedback.

We will close the ticket.

BR

Matthias

#31 - 2023-03-13 09:01 - hidden

- *Project changed from 30 to Public Support*

- *Status changed from To Be Closed to Closed*

- *Private changed from Yes to No*