

Public Support - Support Request #11907

Detection Pintype/-direction

2020-08-12 14:30 - hidden

Status:	Closed	Product Issue Numbers: Affected Products: ADTF 3.8.0 Platform: Windows 10 64bit Topic: ADTF::FilterSDK FAQ Links:
Priority:	Normal	
Category:		
Customer:	AUDI	
Department:	AST	
Requester's Priority:	Normal	
Support Level:	2nd Level	
Resolution:	Solved Issue	

Description

Support Anfrage:

wir lässt sich bei ADTF 3 ermitteln, ob es ein Inputpin oder ein Outputpin ist, wenn ich über this->GetPins() alle Pins des Filters auslese?

Bei ADTF 2 war das über die GetDirection() Methode möglich, aber die gibt es bei ADTF 3 nicht mehr.

Lösung:

Der Editor im CE legt diese Pins nur im CE an, sodass sie im Graphen verwendet werden können. Diese dynamischen Pins werden dann zu Laufzeit über RequestDynamicInputPin() (https://support.digitalwerk.net/adtf/v3/adtf_html/classadtf_1_1filter_1_1flash_1_1c_graph_object.html#a47e3cf5bcffe517ed81f1c285f3b36a1) und RequestDynamicOutputPin() (https://support.digitalwerk.net/adtf/v3/adtf_html/classadtf_1_1filter_1_1flash_1_1c_graph_object.html#a35e74067c9a2c20d99b8fa230c382fa9) beim Filter angefordert und der muss sie in diesen Funktionen dann erstellen. In der Doku sind jeweils Beispiele aufgeführt.

Das bedeutet der Weg über GetPins() ist hier der Falsche :-). Der Vollständigkeit halber: Den Typ eines Pins findet man über einen ucom_cast<> zu IInPin oder IOutputpin raus.

Eigentlich reicht es, wenn ihr euch die Reader/Writer (der Returnwert) von CreateInputPin und CreateOutputPin() in den Request Funktionen merkt (so wie es die Beispiele machen).

Die Methode RequestDynamicPin wird immer für alle Pins des Filter aufgerufen, die nicht statisch erzeugt worden sind oder? Dann wäre der Weg darüber eigentlich besser.

Ja genau!

History

#1 - 2020-08-12 16:31 - hidden

- Project changed from Public Support to 11
- Status changed from New to In Progress
- Customer set to AUDI
- Department set to AST
- Affected Products ADTF 3.8.0 added

#2 - 2020-08-12 16:32 - hidden

- Topic set to ADTF::FilterSDK

#3 - 2020-08-12 17:00 - hidden

Nachtrag:

Die Pins des Filters sind mittels eines QML-Files über createInputPin und createOutputPin angelegt worden. GetPins() liefert aber nur die

Standardpins Trigger_in und Trigger_Out zurück.
Wie lassen sich die per QML angelegten Pins im Filter auslesen?

#4 - 2020-08-13 08:48 - hidden

Hi Dirk,

der Editor im CE legt diese Pins nur im CE an, sodass sie im Graphen verwendet werden können. Diese dynamischen Pins werden dann zu Laufzeit über RequestDynamicInputPin() (https://support.digitalwerk.net/adtf/v3/adtf_html/classadtf_1_1filter_1_1flash_1_1c_graph_object.html#a47e3cf5bcffe517ed81f1c285f3b36a1) und RequestDynamicOutputPin() (https://support.digitalwerk.net/adtf/v3/adtf_html/classadtf_1_1filter_1_1flash_1_1c_graph_object.html#a35e74067c9a2c20d99b8fa230c382fa9) beim Filter angefordert und der muss sie in diesen Funktionen dann erstellen. In der Doku sind jeweils Beispiele aufgeführt.

Das bedeutet der Weg über GetPins() ist hier der Falsche :-). Der Vollständigkeit halber: Den Typ eines Pins findet man über einen ucom_cast<> zu InPin oder IOutputpin raus.

Grüße,

Martin

#5 - 2020-08-13 13:00 - hidden

Danke. Wenn wir die Funktions RequestDynamic Pins implementieren, können wir über GetPins und ucom_cast die angelegten Pins auslesen. Das Ticket kann geschlossen werden.

Viele Grüße

Dirk

#6 - 2020-08-13 13:28 - hidden

Hi Dirk,

nur um sicher zu gehen: Wofür verwendet ihr denn dann die Pins die ihr mit GetPins() holt?

Eigentlich reicht es, wenn ihr euch die Reader/Writer (der Returnwert) von CreateInputPin und CreateOutputPin() in den Request Funktionen merkt (so wie es die Beispiele machen).

Grüße,

Martin

#7 - 2020-08-13 15:15 - hidden

Hallo Martin,

Danke der Nachfrage. Die Filterpins sollen entsprechend einer Konfigurationsdatei, die in den Filterproperties angegeben ist, über das Kontextmenü des Filters und einen QML Skript im CE angelegt werden. Wenn der Filter im Launcher ausgeführt wird, will ich nochmal prüfen, ob die Pins des Filters zu den angegebenen Pins in der Konfigurationsdatei passen. An Reader/Writte hatte ich gar nicht gedacht, aber ich habe gesehen, dass sie auch den Pinnamen enthalten.

Die Methode RequestDynamicPin wird immer für alle Pins des Filter aufgerufen, die nicht statisch erzeugt worden sind oder? Dann wäre der Weg darüber eigentlich besser.

Viele Grüße

Dirk

#8 - 2020-08-17 10:25 - hidden

- Status changed from In Progress to To Be Closed

- Resolution set to Solved Issue

- Platform Windows 10 64bit added

Dirk Koltermann wrote:

Die Methode RequestDynamicPin wird immer für alle Pins des Filter aufgerufen, die nicht statisch erzeugt worden sind oder? Dann wäre der Weg darüber eigentlich besser.

Ja genau!

Grüße,

Martin

#9 - 2020-08-17 12:00 - hidden

Ok. Danke für die Info. Das Ticket kann geschlossen werden.

#10 - 2020-08-18 11:31 - hidden

- *Subject changed from Ermittlung Pintyp/-richtung to Detection Pintype/-direction*

- *Description updated*

#11 - 2020-08-18 11:36 - hidden

- *Project changed from 11 to Public Support*

- *Private changed from Yes to No*

#12 - 2020-10-06 10:27 - hidden

- *Status changed from To Be Closed to Closed*

Files

image001.png	31.2 KB	2020-08-12	hidden
--------------	---------	------------	--------