

Public Support - Support Request #3912

EBPRODUCTSUPPORT-1195 Demo Synchronizer session is missing a Timer Runner that triggers queue processing

2018-08-30 10:30 - hidden

Status:	Closed	Product Issue Numbers:	https://www.cip.audi.de/jira/browse/ACORE-9651 ; https://www.cip.audi.de/jira/browse/ACORE-9672
Priority:	Normal	Affected Products:	ADTF 3.3.3
Category:		Platform:	
Customer:	ELEKTROBIT	Topic:	ADTF::Doc
Department:	SUPPORT	FAQ Links:	
Requester's Priority:	Normal		
Support Level:	3rd Level		
Resolution:	Product Issue Opened		

Description

Supportanfrage

noch ein report von unserem kollegen:

"Die Session "Demo TimeStamp Synchronizer" aus dem ADTF Examples Project "D:\ADTF\3.3.3\src\examples\projects\ADTF_Project\ADTF Project.adtfproject" läuft nicht vernünftig.

Ich hab noch den Sample-Stream-Face-View-UI Service hinzugefügt, um mir das Triggering zu veranschaulichen (wird ja nirgends dokumentiert, was der Filter macht).

Effekt: auf den Sample Streams, die mit den Ausgabe-Pins des Demo Synchronizer Filters verbunden sind, kommt nichts an. Siehe Attachment TraceView.PNG

Dagegen läuft nach kurzer Zeit der Memory Pool voll. (Attachment Log.PNG).

An den Einstellungen habe ich sonst nichts gemacht. Die Binaries sind die Originalen von der Installation.

Das Beispiel sollte so modifiziert werden, dass die Trigger auf den Ausgabepins funktionieren, bzw. im Trace View sichtbar sind, und der Speicher nicht mit den gegebenen Einstellungen überläuft."

Jochen vermutete erst ein problem mit den triggern:

"Nun steht allerdings in der Doku von `adtf::streaming::ant::cSampleWriter::ManualTrigger` (das ist die Methode die Aufgerufen wird) folgende Warning:

Zeile 158 in `samplewriter.h`

- @warning do not call this within a filter! You will destroy the runtimebehaviour provided by a the @ref subsection_trigger_pipe. Meine Frage dazu an die Entwickler, oder einen Experten von EB wäre:

1. Ist die Doku hier fehlerhaft, wie so oft? Wenn ja, als Bug-Ticket für die Doku aufnehmen. Oder ist es ein Fehler im Example?
2. Wenn kein Doku-Fehler nein: Welchen schlimmen Effekt hätte es, wenn man triggering im Filter nicht über die Innerpipes macht, sondern manuelle Trigger verwendet. Ein Vorteil der manuellen trigger ist immerhin, dass man die gezielt schicken kann (z.B. bei jedem sample transmit). Während bei den triggerpipes auch ein trigger geschickt wird, wenn der runner kein sample rausschickt (z.B. weil er mit Fehler abbricht)."

Lösung

1. Ist die Doku hier fehlerhaft, wie so oft? Wenn ja, als Bug-Ticket für die Doku aufnehmen. Oder ist es ein Fehler im Example?

Also was der Filter macht ist in der Doku genau beschrieben -> [Synchronizer Filter](#)

Problem ist, dass in der Beispielkonfiguration die Property `synchronous_queue_processing` nicht aktiviert ist (oder ein eigener Timer Runner fehlt). Dadurch kommt der Filter nie dazu seine queues zu überprüfen und sie laufen voll. Die Property ist in obiger Doku auch erwähnt. -> ACORE-9651 erstellt.

Die Doku ist an dieser und vielen anderen Stellen nicht fehlerhaft.

2. Wenn kein Doku-Fehler nein: Welchen schlimmen Effekt hätte es, wenn man triggering im Filter nicht über die Innerpipes macht, sondern manuelle Trigger verwendet.
Ein Vorteil der manuellen trigger ist immerhin, dass man die gezielt schicken kann (z.B. bei jedem sample transmit). Während bei den triggerpipes auch ein trigger geschickt wird, wenn der runner kein sample rausschickt (z.B. weil er mit Fehler abbricht)."

Das Problem an den Manual Triggern ist, dass sie nicht konfigurierbar sind. Ziel in ADTF3 ist es in Zukunft auch die Inner Pipes über den CE konfigurierbar zu machen. Das heißt das Laufzeitverhalten ist komplett von außen nachvollzieh- und konfigurierbar.

Wenn kein Sample verschickt worden ist, sollte auch kein Trigger ausgelöst werden (im Fehlerfall greift hier in Zukunft ein zentrales Fehlerhandling in ADTF über SetStreamError der Reader und Writer), aber selbst in dem Fall ist es kein Problem wenn die restliche Triggerkette leer durchläuft. Auf den Fall dass eine Reader kein Sample liefert, müssen alle Triggerfunktionen etc. vorbereitet sein (und das sind unsere Beispiele auf jeden Fall).

Bezüglich dem manuellen Trigger. Wenn ich Dich richtig verstanden habe, muss man die Info aus der Doku

```
"@warning do not call this within a filter! You will destroy the runtimebehaviour provided by a the @ref subsection_trigger_pipe."
```

so interpretieren, dass man manuelle Trigger nur senden soll, wenn man die gewünschte Funktionalität nicht anders erreichen kann. Und sonst die Trigger Pipes zur Konfiguration nehmen soll.

Ja.

Ich denke da z.B. an Filter wie den CAN Config Codec aus ADTF 2, wo kein einfaches Verhältnis zwischen empfangenen input-samples und gesendeten output-samples besteht. Oder einen Gui-Filter der samples aus dem Qt-thread sendet (so wie das ADTF 2 Qt-Dialog-Display). Es ist hier also auch ok, wenn der Demo Synchronizer manuell Trigger sendet.

Ja, das macht er ja auch, aber er leitet die trigger genauso verzögert weiter wie die Samples.

(Oder war der Plan, dass solche Funktionalität in Filtern nicht mehr unterstützt werden soll, und nur noch ganz einfache Verarbeitungsverhältnisse (ein input trigger, ein output trigger, oder eben threads, timer filter, die bei jedem Aufruf genau ein neues sample produzieren, auch wenn auf dem Input währenddessen keins oder viele samples reinkamen)

Nein, natürlich sollen komplexe Szenarien nach wie vor möglich sein, genau dafür gibts ManualTrigger() oder besser adtf::streaming::trigger. Es wäre vermessen anzunehmen, ein Config Tool zu schaffen, dass z.b. die von dir erwähnten Abhängigkeiten innerhalb eines CCC modelliert. Für reine Regulations etc. Filter, die die Mehrzahl der von "Durchschnittsusern" erstellten Filtern (oder besser Triggerfunktionen) darstellen dürften, soll es aber eben nicht hart kodiert sein.

Wenn ersteres zutrifft, klingt allerdings das Warning für mich etwas missverständlich (heißt ja wörtlich, "mach es nicht"), und ich würde vorschlagen dass man das dann etwas klarer formuliert.

Ja, da könnte noch ein Zusatz dazu, dass das an der Stelle dann einen synchronen Aufruf der nachgelagerten runnerpipes darstellt, der von außen nicht konfigurierbar und ersichtlich ist, und man das nur machen soll wenn mans nicht anders modelieren kann -> ACORE-9672 erstellt.

History

#1 - 2018-08-30 11:07 - hidden

- Project changed from Public Support to 7
- Topic set to ADTF::FilterSDK
- Customer set to ELEKTROBIT
- Department set to SUPPORT
- Affected Products ADTF 3.3.3 added

#2 - 2018-09-03 09:18 - hidden

- Status changed from New to In Progress

Hi,

also was der Filter macht ist in der Doku genau beschrieben: https://support.digitalwerk.net/adtf/v3/adtf_html/page_example_synchronizer_filter.html

Problem ist, dass in der Beispielkonfiguration die Property `synchronous_queue_processing` nicht aktiviert ist (oder ein eigener Timer Runner fehlt). Dadurch kommt der Filter nie dazu seine queues zu überprüfen und sie laufen voll. Die Property ist in obiger Doku auch erwähnt. Dafür erstell ich ein Ticket.

Die Doku ist an dieser und vielen anderen Stellen nicht fehlerhaft:

zu 2.:

Das Problem an den Manual Triggern ist, dass sie nicht konfigurierbar sind. Ziel in ADTF3 ist es in Zukunft auch die Inner Pipes über den CE konfigurierbar zu machen. Das heißt das Laufzeitverhalten ist komplett von außen nachvollzieh- und konfigurierbar.

Wenn kein Sample verschickt worden ist, sollte auch kein Trigger ausgelöst werden (im Fehlerfall greift hier in Zukunft ein zentrales Fehlerhandling in ADTF über `SetStreamError` der Reader und Writer), aber selbst in dem Fall ist es kein Problem wenn die restliche Triggerkette leer durchläuft. Auf den Fall dass eine Reader kein Sample liefert, müssen alle Triggerfunktionen etc. vorbereitet sein (und das sind unsere Beispiele auf jeden Fall).

Grüße,

Martin

#3 - 2018-09-03 09:26 - hidden

Ich habe Ticket ACORE-9651 erstellt.

#4 - 2018-09-03 09:54 - hidden

- Subject changed from *EBPRODUCTSUPPORT-1195 ADTF 3.3.3 Examples: Demo Synchronizer Filter* to *EBPRODUCTSUPPORT-1195 Demo Synchronizer session is missing a Timer Runner that triggers queue processing*

- Status changed from *In Progress* to *Customer Feedback Required*

- Topic changed from *ADTF::FilterSDK* to *ADTF::Doc*

- Resolution set to *Product Issue Opened*

- Product Issue Numbers set to <https://www.cip.audi.de/jira/browse/ACORE-9651>

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

#6 - 2018-09-10 09:45 - hidden

Noch ein Nachtrag von Jochen:

Hallo Martin,

ja, nach setzen der Property läuft das Beispielprojekt.

Bezüglich dem manuellen Trigger. Wenn ich Dich richtig verstanden habe, muss man die Info aus der Doku

"@warning do not call this within a filter! You will destroy the runtimebehaviour provided by a the @ref subsection_trigger_pipe."

so interpretieren, dass man manuelle Trigger nur senden soll, wenn man die gewünschte Funktionalität nicht anders erreichen kann. Und sonst die Trigger Pipes zur Konfiguration nehmen soll.

Ich denke da z.B. an Filter wie den CAN Config Codec aus ADTF 2, wo kein einfaches Verhältnis zwischen empfangenen input-samples und gesendeten output-samples besteht. Oder einen Gui-Filter der samples aus dem Qt-thread sendet (so wie das ADTF 2 Qt-Dialog-Display). Es ist hier also auch ok, wenn der Demo Synchronzier manuell Trigger sendet.

(Oder war der Plan, dass solche Funktionalität in Filtern nicht mehr unterstützt werden soll, und nur noch ganz einfache Verarbeitungsverhältnisse (ein input trigger, ein output trigger, oder eben threads, timer filter, die bei jedem Aufruf genau ein neues sample produzieren, auch wenn auf dem Input währenddessen keins oder viele samples reinkamen)

Wenn ersteres zutrifft, klingt allerdings das Warning für mich etwas missverständlich (heißt ja wörtlich, "mach es nicht"), und ich würde vorschlagen dass man das dann etwas klarer formuliert.

Best regards - Beste Grüße,

Florian Obermeier

EB Assist ADTF Support-Team

#7 - 2018-09-10 15:15 - hidden

- Status changed from *Customer Feedback Required* to *In Progress*

#9 - 2018-09-11 14:23 - hidden

Bezüglich dem manuellen Trigger. Wenn ich Dich richtig verstanden habe, muss man die Info aus der Doku

"@warning do not call this within a filter! You will destroy the runtimebehaviour provided by a the @ref subsection_trigger_pipe."

so interpretieren, dass man manuelle Trigger nur senden soll, wenn man die gewünschte Funktionalität nicht anders erreichen kann. Und sonst die Trigger Pipes zur Konfiguration nehmen soll.

Ja.

Ich denke da z.B. an Filter wie den CAN Config Codec aus ADTF 2, wo kein einfaches Verhältnis zwischen empfangenen input-samples und gesendeten output-samples besteht. Oder einen Gui-Filter der samples aus dem Qt-thread sendet (so wie das ADTF 2 Qt-Dialog-Display). Es ist hier also auch ok, wenn der Demo Synchronizer manuell Trigger sendet.

Ja, das macht er ja auch, aber er leitet die trigger genauso verzögert weiter wie die Samples.

(Oder war der Plan, dass solche Funktionalität in Filtern nicht mehr unterstützt werden soll, und nur noch ganz einfache Verarbeitungsverhältnisse (ein input trigger, ein output trigger, oder eben threads, timer filter, die bei jedem Aufruf genau ein neues sample produzieren, auch wenn auf dem Input währenddessen keins oder viele samples reinkamen)

Nein, natürlich sollen komplexe Szenarien nach wie vor möglich sein, genau dafür gibts ManualTrigger() oder besser adtf::streaming::trigger. Es wäre vermessen anzunehmen, ein Config Tool zu schaffen, dass z.b. die von dir erwähnten Abhängigkeiten innerhalb eines CCC modelliert. Für reine Regelungen etc. Filter, die die Mehrzahl der von "Durchschnittsusern" erstellten Filtern (oder besser Triggerfunktionen) darstellen dürften, soll es aber eben nicht hart kodiert sein.

Wenn ersteres zutrifft, klingt allerdings das Warning für mich etwas missverständlich (heißt ja wörtlich, "mach es nicht"), und ich würde vorschlagen dass man das dann etwas klarer formuliert.

Ja, da könnte noch ein Zusatz dazu, dass das an der Stelle dann einen synchronen Aufruf der nachgelagerten runnerpipes darstellt, der von außen nicht konfigurierbar und ersichtlich ist, und man das nur machen soll wenn mans nicht anders modellieren kann. Ich mach Euch ein Ticket...

#10 - 2018-09-11 14:27 - hidden

Ticket ist ACORE-9672.

#11 - 2018-09-12 17:35 - hidden

- Status changed from In Progress to Customer Feedback Required

- Product Issue Numbers changed from <https://www.cip.audi.de/jira/browse/ACORE-9651> to <https://www.cip.audi.de/jira/browse/ACORE-9651>; <https://www.cip.audi.de/jira/browse/ACORE-9672>

#12 - 2018-09-17 09:19 - hidden

- Project changed from 7 to Public Support

- Description updated

- Status changed from Customer Feedback Required to To Be Closed

- Private changed from Yes to No

#13 - 2018-09-17 09:20 - hidden

- Status changed from To Be Closed to Closed

Files

Log.PNG	129 KB	2018-08-30	hidden
TraceView.PNG	44.7 KB	2018-08-30	hidden