Public Support - Support Request #2431

Streaming Source/Sink to read and write over the same TCP connection

2018-04-30 08:40 - hidden

Status:	Closed		
Priority:	Normal		
Category:			
Customer:	VW	**Product Issue Numbers:**	
Department:	CARMEQ	**Affected Products:**	ADTF 3.3.0
Requester's Priority:	Normal	**Platform:**	Ubuntu 16.04 64bit, Windows 10 64bit
Support Level:	2nd Level	**Topic:**	ADTF::IPC
Resolution:	Solved Issue	**FAQ Links:**	

Description

Supportanfrage

Ich bin dabei, eine Streaming Source zu implementieren, die eine TCP-Verbindung aufbaut und die empfangenen Daten als Mediasamles streamed (siehe issue #2390). Nun möchte ich aber zusätzlich noch Daten aus der ADTF-Konfiguration *über die bestehende TCP-Verbindung* versenden.

Es würden sich m.E. mehrere Möglichkeiten anbieten:

1. Einen kombinierten Filter schreiben, der eine TCP-Verbindung öffnet und gleichzeitig als Streaming-Source und -Sink fungiert.
   -> Ist das überhaupt vorgesehen? Wie würde man da vorgehen?
2. Einen Hintergrund-Service schreiben, der die TCP-Verbindung öffnet und gleichzeitig, Daten von speziellen Sinks und Sources empfängt und über die Verbindung leitet.
   -> Scheint sehr aufwändig und durch die impliziten Verbindungen nicht besonders intuitiv.
3. Der Receiver (StreamingSource) macht das TCP Connection Handling und ein separater Sender (Streaming-Sink) nutzt Funktionen des Receivers um Daten über dessen offene Verbindung zu senden.
   -> Wie würde der Sender seinen dazugehörigen Receiver erkennen? Wäre hier der *Interface-Pin* Mechanismus hilfreich? Wie setzt man das um? Gibt es da passende Beispiele?

Wie ist hierfür das empfohlene Vorgehen in ADTF3?

Lösung

Variante 3:

Bitte einfach genau das foreign_application_udp Beispiel ansehen. Dort wird genau das über ein Interface Binding gemacht. Heißt wenn Source und Sink
verbunden sind, verwenden sie ein und denselben Socket, ansonsten Eigenständige.

History

#2 - 2018-04-30 09:07 - hidden

- Topic set to ADTF::IPC

#3 - 2018-05-03 09:08 - hidden

- Status changed from New to In Progress

Variante 3:

Bitte einfach genau das foreign_application_udp Beispiel ansehen. Dort wird genau das über ein Interface Binding gemacht. Heißt wenn Source und Sink
verbunden sind, verwenden sie ein und denselben Socket, ansonsten Eigenständige.

#4 - 2018-05-03 11:24 - hidden

- Status changed from In Progress to Customer Feedback Required

#5 - 2018-05-03 12:29 - hidden

Kann geschlossen werden.

#6 - 2018-05-03 13:20 - hidden

- Description updated

- Status changed from Customer Feedback Required to To Be Closed

- Resolution set to Solved Issue

#7 - 2018-05-22 08:15 - hidden

- Project changed from 20 to Public Support

- Private changed from Yes to No

#8 - 2018-05-23 13:36 - hidden

- Status changed from To Be Closed to Closed