Public Support - Support Request #8617

Seperate adtf 3 setup in a modular way

2019-10-17 12:56 - hidden

Status:	Closed		
Priority:	Normal		
Category:			
Customer:	AUDI	**Product Issue Numbers:**	
Department:	EF	**Affected Products:**	ADTF 3.6.2
Requester's Priority:	Normal	**Platform:**	Ubuntu 18.04 64bit, Windows 10 64bit
Support Level:	2nd Level	**Topic:**	ADTF::Common
Resolution:	Solved Issue	**FAQ Links:**	

Description

Supportanfrage

Zur Historie:

Mit ADTF 2 nutzen wir eine große Haupt-Config "system.xml", in die kleinere bis mittlere Configs, die von Zulieferern kommen, reinintegriert werden. Hierzu hatten wir das inoffizielle "<include>"-Tag verwendet. Zusätzlich gab es ein Tool, welches ggf. Anpassungen an den Inport/Outport-Schnittstellen oder Pfaden der zugelieferten Sub-Config durchgeführt hat.

Vorhaben:

Kann man mit ADTF 3 ähnlich modular verfahren? Zusätzlich ist es mit ADTF 3 ja möglich, mehrere Sessions auf verschiedenen Ports miteinander kommunizieren zu lassen. Wäre das auch für größere Mengen an Modulen gangbar oder gar zu bevorzugen -- verglichen mit einer großen Config mit vielen "<include>"-Anweisungen?

Wie sieht es mit der Nutzung von Umgebungsvariablen (bspw. für Pfade oder Property-Werte) in den ADTF3-Config-Files aus?

Lösung

leider gibt es in ADTF3 noch keine Entsprechung zum Include aus ADTF2. Es gibt dafür ein Ticket ACORE-8169, aber das ist leider noch nicht umgesetzt.

Ja das Auslagern in eigene Sessions die via IPC miteinander kommunizieren ist eine gute Möglichkeit diesen Anwendungsfall jetzt abzudecken. Bringt aber ansonsten keine Vorteile gegenüber einer einzelnen Session und erhöht die Komplexität (z.b. beim Starten).

Ja Umgebungsvariablen funktionieren in Properties und auch in Pfaden zu z.b. Plugins (.adtfsystem).

> Die Config-Netzwerk-via-IPC-Lösung hätte aus meiner Sicht den Reiz, daß die einzelnen Modul-Lieferanten ihre Sub-Configs liefern relativ schnell klar wird, welche Config nicht funktioniert, bspw. nicht einmal in den Init-Zustand kommt oder crasht. Wir selbst müßten nur die IPC-Schnittstellen festlegen.

Das auf jeden Fall!

> Wie sähe bei einem solchen Config-Verbund die Steuerung aus? Ich müßte mich um eine "Kommandozentrale" kümmern, welche den Session-Verbund startet, stoppt und überwacht. Müßte ich das mit der Control-CLI machen? Oder gibt es auch eine mitgelieferte C++-Lib dazu?

Hier haben wir leider keine Out-of-the-box Lösung. Die von dir angedeuteten Wege gibt es aber beide:

- Scripting via adtf_control. Dazu haben wir Beispiel Skripte (Shell und Python) in der ADTF Installation (https://support.digitalwerk.net/adtf/v3/adtf_html/page_demo_script_control_playback.html)
- Zugriff via C++ API (https://support.digitalwerk.net/adtf/v3/adtf_html/mainpage_remotesdk_pkg.html)

Zusätzlich kann man auch das ADTF GUI Control mit mehreren Sessions verbinden, da gibt es aber noch keine Funktionalität sie

"synchron" zu starten oder stoppen.

Für die Zukunft ist aber angedacht, dass Management eines Verteilten ADTF Systems über FEP zu Lösen. @Sebastian kannst Du da bitte kurz etwas dazu erleutern?

> Thema Time-Sync bei dieser Lösung: Kann man wie damals beim MessageBus aus ADTF2 wieder eine Master-Session für die Uhr festlegen, um alle in den Takt zu bringen? Kann man alle Sessions synchron starten?

Ja das funktioniert, dazu muss man die "Slave" Sessions mit dem ADTF Clock Synchronization Service ausstatten und den dann an den Master verweisen: https://support.digitalwerk.net/adtf/v3/adtf_html/page_clock_synchronization_service_plugin.html

Zeithorizont für das "Include"-Feature kann ich leider nicht wirklich nennen, aber vor Mitte nächsten Jahres würde ich nicht damit rechnen (das zieht leider sehr sehr viel nach sich, sodass das eine ziemlich "großes" Feature ist).

> In diesem Fall hätte ich auch das synchrone Starten des Session-Verbunds erschlagen, richtig? Ich stelle mir einen Master vor, der alle Slaves mitstartet (wie bei ADTF2 mit dem Command Channel, oder wie das hieß).

Nein, das Starten übernimmt nicht der Zeitmaster, sondern muss von außen sichergestellt werden. Die ADTF3 Instanzen steuern sich nie gegenseitig. Die "Slaves" synchronisieren ihre Zeit beim Wechsel in Runlevel Running, heißt man muss "nur" sicherstellen, dass der Master als erster gestartet wird.

> Bei der ganzen Sache interessiert uns auch die Plugin-Description, da die Modul-Lieferanten somit die Abhängigkeiten der Plugins dokumentieren müssen. Können hierbei neben ADTF-Services und ADTF-Interfaces auch externe DLLs oder Shared Objects angegeben werden, die beim Laden des Plugins zwingend verfügbar sein müssen? Ich habe dazu in der Doku einen Hinweis gefunden. Was ist, wenn ein Filter eine DDL per "LoadLibrary" zur Laufzeit hinzulädt? Würde man diese trotzdem in die Plugin-Description eintragen?

Ja es können dort unter <platform_dependencies> auch externe Abhängigkeiten angegeben werden. Ein Beispiel findet man unter https://support.digitalwerk.net/adtf/v3/adtf_html/page_plugin_description.html. Diese werden vom adtf_launcher dann automatisch beim Start (per LoadLibrary) geladen bevor das Plugin geladen wird. Wenn das Filter Plugin die Library selbst auch erst mit LoadLibrary lädt und nicht direkt dagegen gelinkt ist, kann man diese Abhängigkeit ruhig auch eintragen, dann hat man mehr Möglichkeiten den Pfad anzupassen (falls die Abhängigkeit nicht in PATH ist oder direkt neben der Filter PLB liegt). Kommt darauf an wie der Filter seinen Pfad für LoadLibrary zusammenbastelt.

History

#1 - 2019-10-23 08:41 - hidden

- Status changed from New to In Progress

- Topic set to ADTF::Common

@Martin Heimlich bitte anschauen

#2 - 2019-10-28 08:28 - hidden

Hallo,

leider gibt es in ADTF3 noch keine Entsprechung zum Include aus ADTF2. Es gibt dafür ein Ticket ACORE-8169, aber das ist leider noch nicht umgesetzt.

Ja das Auslagern in eigene Sessions die via IPC miteinander kommunizieren ist eine gute Möglichkeit diesen Anwendungsfall jetzt abzudecken. Bringt aber ansonsten keine Vorteile gegenüber einer einzelnen Session und erhöht die Komplexität (z.b. beim Starten).

Ja Umgebungsvariablen funktionieren in Properties und auch in Pfaden zu z.b. Plugins (.adtfsystem).

Beste Grüße,

Martin

#3 - 2019-10-28 08:45 - hidden

Hallo Martin,

gibt es einen groben, unverbindlichen Zeithorizont für das Include-Feature, bspw. ADTF 3.x oder so?

Die Config-Netzwerk-via-IPC-Lösung hätte aus meiner Sicht den Reiz, daß die einzelnen Modul-Lieferanten ihre Sub-Configs liefern relativ schnell klar wird, welche Config nicht funktioniert, bspw. nicht einmal in den Init-Zustand kommt oder crasht. Wir selbst müßten nur die IPC-Schnittstellen

festlegen.

Wie sähe bei einem solchen Config-Verbund die Steuerung aus? Ich müßte mich um eine "Kommandozentrale" kümmern, welche den Session-Verbund startet, stoppt und überwacht. Müßte ich das mit der Control-CLI machen? Oder gibt es auch eine mitgelieferte C++-Lib dazu?

Thema Time-Sync bei dieser Lösung: Kann man wie damals beim MessageBus aus ADTF2 wieder eine Master-Session für die Uhr festlegen, um alle in den Takt zu bringen? Kann man alle Sessions synchron starten?

#4 - 2019-10-28 09:16 - hidden

Hi Patrick

> Die Config-Netzwerk-via-IPC-Lösung hätte aus meiner Sicht den Reiz, daß die einzelnen Modul-Lieferanten ihre Sub-Configs liefern relativ schnell klar wird, welche Config nicht funktioniert, bspw. nicht einmal in den Init-Zustand kommt oder crasht. Wir selbst müßten nur die IPC-Schnittstellen festlegen.

Das auf jeden Fall!

> Wie sähe bei einem solchen Config-Verbund die Steuerung aus? Ich müßte mich um eine "Kommandozentrale" kümmern, welche den Session-Verbund startet, stoppt und überwacht. Müßte ich das mit der Control-CLI machen? Oder gibt es auch eine mitgelieferte C++-Lib dazu?

Hier haben wir leider keine Out-of-the-box Lösung. Die von dir angedeuteten Wege gibt es aber beide:

- Scripting via adtf_control. Dazu haben wir Beispiel Skripte (Shell und Python) in der ADTF Installation (https://support.digitalwerk.net/adtf/v3/adtf_html/page_demo_script_control_playback.html)
- Zugriff via C++ API (https://support.digitalwerk.net/adtf/v3/adtf_html/mainpage_remotesdk_pkg.html)

Zusätzlich kann man auch das ADTF GUI Control mit mehreren Sessions verbinden, da gibt es aber noch keine Funktionalität sie "synchron" zu starten oder stoppen.

Für die Zukunft ist aber angedacht, dass Management eines Verteilten ADTF Systems über FEP zu Lösen. @Sebastian kannst Du da bitte kurz etwas dazu erleutern?

> Thema Time-Sync bei dieser Lösung: Kann man wie damals beim MessageBus aus ADTF2 wieder eine Master-Session für die Uhr festlegen, um alle in den Takt zu bringen? Kann man alle Sessions synchron starten?

Ja das funktioniert, dazu muss man die "Slave" Sessions mit dem ADTF Clock Synchronization Service ausstatten und den dann an den Master verweisen: https://support.digitalwerk.net/adtf/v3/adtf_html/page_clock_synchronization_service_plugin.html

Zeithorizont für das "Include"-Feature kann ich leider nicht wirklich nennen, aber vor Mitte nächsten Jahres würde ich nicht damit rechnen (das zieht leider sehr sehr viel nach sich, sodass das eine ziemlich "großes" Feature ist).

#5 - 2019-10-28 13:13 - hidden

Ja das funktioniert, dazu muss man die "Slave" Sessions mit dem ADTF Clock Synchronization Service ausstatten und den dann an den Master verweisen: https://support.digitalwerk.net/adtf/v3/adtf_html/page_clock_synchronization_service_plugin.html
In diesem Fall hätte ich auch das synchrone Starten des Session-Verbunds erschlagen, richtig? Ich stelle mir einen Master vor, der alle Slaves mitstartet (wie bei ADTF2 mit dem Command Channel, oder wie das hieß).

Bei der ganzen Sache interessiert uns auch die Plugin-Description, da die Modul-Lieferanten somit die Abhängigkeiten der Plugins dokumentieren müssen. Können hierbei neben ADTF-Services und ADTF-Interfaces auch externe DLLs oder Shared Objects angegeben werden, die beim Laden des Plugins zwingend verfügbar sein müssen? Ich habe dazu in der Doku einen Hinweis gefunden. Was ist, wenn ein Filter eine DDL per "LoadLibrary" zur Laufzeit hinzulädt? Würde man diese trotzdem in die Plugin-Description eintragen?

#7 - 2019-11-04 08:38 - hidden

> In diesem Fall hätte ich auch das synchrone Starten des Session-Verbunds erschlagen, richtig? Ich stelle mir einen Master vor, der alle Slaves mitstartet (wie bei ADTF2 mit dem Command Channel, oder wie das hieß).

Nein, das Starten übernimmt nicht der Zeitmaster, sondern muss von außen sichergestellt werden. Die ADTF3 Instanzen steuern sich nie gegenseitig. Die "Slaves" synchronisieren ihre Zeit beim Wechsel in Runlevel Running, heißt man muss "nur" sicherstellen, dass der Master als erster gestartet wird.

> Bei der ganzen Sache interessiert uns auch die Plugin-Description, da die Modul-Lieferanten somit die Abhängigkeiten der Plugins dokumentieren müssen. Können hierbei neben ADTF-Services und ADTF-Interfaces auch externe DLLs oder Shared Objects angegeben werden, die beim Laden des Plugins zwingend verfügbar sein müssen? Ich habe dazu in der Doku einen Hinweis gefunden. Was ist, wenn ein Filter eine DDL per "LoadLibrary" zur Laufzeit hinzulädt? Würde man diese trotzdem in die Plugin-Description eintragen?

Ja es können dort unter <platform_dependencies> auch externe Abhängigkeiten angegeben werden. Ein Beispiel findet man unter https://support.digitalwerk.net/adtf/v3/adtf_html/page_plugin_description.html. Diese werden vom adtf_launcher dann automatisch beim Start (per LoadLibrary) geladen bevor das Plugin geladen wird. Wenn das Filter Plugin die Library selbst auch erst mit LoadLibrary lädt und nicht direkt dagegen gelinkt ist, kann man diese Abhängigkeit ruhig auch eintragen, dann hat man mehr Möglichkeiten den Pfad anzupassen (falls die Abhängigkeit nicht in PATH ist oder direkt neben der Filter PLB liegt). Kommt darauf an wie der Filter seinen Pfad für LoadLibrary zusammenbastelt.

#8 - 2019-11-14 08:34 - hidden

- Status changed from In Progress to Customer Feedback Required

Hallo Patrick,

hilft dir das weiter ?

#10 - 2019-11-18 17:15 - hidden

- Subject changed from Config-Modularisierung mit ADTF 3 to Seperate adtf 3 setup in a modular way

- Description updated

- Status changed from Customer Feedback Required to To Be Closed

- Private changed from Yes to No

- Resolution set to Solved Issue

#11 - 2020-07-07 12:45 - hidden

- Status changed from To Be Closed to Closed