

## Public Support - Support Request #12113

### Design techniques for stability behavior in ADTF 3

2020-09-16 11:46 - hidden

<b>Status:</b>	Closed	<b>Product Issue Numbers:</b>  <b>Affected Products:</b> ADTF 3.9.0 <b>Platform:</b> Windows 10 64bit <b>Topic:</b> ADTF::Common <b>FAQ Links:</b>
<b>Priority:</b>	Normal	
<b>Category:</b>		
<b>Customer:</b>	BOSCH	
<b>Department:</b>	CC-DA/ETV	
<b>Requester's Priority:</b>	Low	
<b>Support Level:</b>	2nd Level	
<b>Resolution:</b>	No Customer Feedback	
<b>Description</b>		
<b>Support Anfrage:</b>		
<p>Aktuell arbeiten wir mit ADTF 2. Hauptsächlich integrieren wir Filter in ADTF Konfigurationen, die von anderen Teams entwickelt werden.</p> <p>Dadurch haben wir keinen direkten Einfluss auf die Entwicklung der einzelnen Filter. In ADTF 2 ist es generell so, dass ein abstürzender Filter zum Absturz des kompletten ADTF Prozesses führt. Gibt es in ADTF 3 eine Möglichkeit auf der Integrationsebene von ADTF Konfigurationen und Filtern dies zu gestalten, deshalb meine Frage dazu:</p> <p>Welche Designtechniken in ADTF 3 können angewendet werden um Absturzverhalten von Filtern einzugrenzen?</p> <p>Zum Beispiel soll ein Absturz in einem Visualisierungsfilter nicht zum Absturz der Aufzeichnung (Aufzeichnungsfilters) führen.</p> <p>Meine Überlegung dazu :</p> <p>Ist die Möglichkeit vorgesehen ADTF Prozesses z.B. über Outports / Inports miteinander kommunizieren (über Interprozesskommunikation) zu lassen ?</p> <p>So könnte man den Visualisierungsfilter in einem anderen ADTF Prozess laufen lassen als den Aufnahmefilter.</p> <p>Ein Absturz des Visualisierungsfilter würd dann nicht mehr zum Absturz der Aufnahme führen.</p>		
<b>Lösung:</b>		
<p>Prinzipiell verhält sich hier ADTF3 wie ADTF2, das heißt die Filter die in einem ADTF Prozess laufen, können auch den ganzen Prozess abschießen.</p> <p>Aber im Gegensatz zu ADTF2 gibt es auch einfach verwendbare IPC Mechanismen (z.b. <a href="https://support.digitalwerk.net/adtf/v3/adtf_html/page_ipc_host_only_plugin.html#section_host_only_ipc_sink">https://support.digitalwerk.net/adtf/v3/adtf_html/page_ipc_host_only_plugin.html#section_host_only_ipc_sink</a> und <a href="https://support.digitalwerk.net/adtf/v3/adtf_html/page_ipc_host_only_plugin.html#section_host_only_ipc_source">https://support.digitalwerk.net/adtf/v3/adtf_html/page_ipc_host_only_plugin.html#section_host_only_ipc_source</a>) mit denen das von dir beschriebene Szenario mit der Aufteilung in mehrere Prozesse recht einfach möglich ist. Eine Beispiel Session findest Du unter <a href="https://support.digitalwerk.net/adtf/v3/adtf_html/page_demo_ipc_sessions.html">https://support.digitalwerk.net/adtf/v3/adtf_html/page_demo_ipc_sessions.html</a></p> <p>Bleibt nur noch das Problem des Managements der ADTF Prozesse (starten/stoppen etc.). Hier gibt es leider keine Komplettlösung, aber da bei ADTF 3 jeder adtf_launcher Prozess sowieso nur über RPC gesteuert wird, kann man das selbst, z.b. mit dem adtf_control handhaben und an den eigenen Use-Case anpassen.</p> <p>Was in Zukunft die Lösung für so ein verteiltes System sein wird, ist ADTF mit FEP integration ( <a href="https://support.digitalwerk.net/adtf/v3_fep_integration_preview/guides/fep3_integration_prologue.html">https://support.digitalwerk.net/adtf/v3_fep_integration_preview/guides/fep3_integration_prologue.html</a>). Hier können dann mehrere ADTF Instanzen als FEP Participants gemeinsam in einem FEP System gesteuert werden und auch über FEP Daten austauschen. Das ist zwar schon weit gediehen, aber leider noch nicht öffentlich verfügbar.</p>		

#### History

#1 - 2020-09-17 10:20 - hidden

- Status changed from New to In Progress
- Topic set to ADTF::Common

#2 - 2020-09-18 08:50 - hidden

Hi Thomas,

prinzipiell verhält sich hier ADTF3 wie ADTF2, das heißt die Filter die in einem ADTF Prozess laufen, können auch den ganzen Prozess abschließen.

Aber im Gegensatz zu ADTF2 gibt es auch einfach verwendbare IPC Mechanismen (z.b.

[https://support.digitalwerk.net/adtf/v3/adtf\\_html/page\\_ipc\\_host\\_only\\_plugin.html#section\\_host\\_only\\_ipc\\_sink](https://support.digitalwerk.net/adtf/v3/adtf_html/page_ipc_host_only_plugin.html#section_host_only_ipc_sink) und [https://support.digitalwerk.net/adtf/v3/adtf\\_html/page\\_ipc\\_host\\_only\\_plugin.html#section\\_host\\_only\\_ipc\\_source](https://support.digitalwerk.net/adtf/v3/adtf_html/page_ipc_host_only_plugin.html#section_host_only_ipc_source)) mit denen das von dir beschriebene Szenario mit der Aufteilung in mehrere Prozesse recht einfach möglich ist. Eine Beispiel Session findest Du unter [https://support.digitalwerk.net/adtf/v3/adtf\\_html/page\\_demo\\_ipc\\_sessions.html](https://support.digitalwerk.net/adtf/v3/adtf_html/page_demo_ipc_sessions.html)

Bleibt nur noch das Problem des Managements der ADTF Prozesse (starten/stoppen etc.). Hier gibt es leider keine Komplettlösung, aber da bei ADTF 3 jeder adtf\_launcher Prozess sowieso nur über RPC gesteuert wird, kann man das selbst, z.b. mit dem adtf\_control handhaben und an den eigenen Use-Case anpassen.

Was in Zukunft die Lösung für so ein verteiltes System sein wird, ist ADTF mit FEP integration ([https://support.digitalwerk.net/adtf/v3\\_fep\\_integration\\_preview/guides/fep3\\_integration\\_prologue.html](https://support.digitalwerk.net/adtf/v3_fep_integration_preview/guides/fep3_integration_prologue.html)). Hier können dann mehrere ADTF Instanzen als FEP Participants gemeinsam in einem FEP System gesteuert werden und auch über FEP Daten austauschen. Das ist zwar schon weit gediehen, aber leider noch nicht öffentlich verfügbar.

Ich hoffe das hilft Dir fürs erste, bitte melde Dich mit weiteren Fragen gerne wieder!

Grüße,

Martin

### **#3 - 2020-09-30 09:29 - hidden**

- Status changed from In Progress to To Be Closed
- Resolution set to No Customer Feedback
- Platform Windows 10 64bit added

### **#4 - 2020-09-30 13:10 - hidden**

- Subject changed from Desgintechiken für Stabilitätsverhalten in ADTF 3 to Design techniques for stability behavior in ADTF 3
- Description updated

### **#5 - 2020-09-30 13:36 - hidden**

- Project changed from 5 to Public Support
- Status changed from To Be Closed to Closed
- Private changed from Yes to No

Vor dem Schließen möchte ich noch ergänzen, dass wir im ADTF 3 (Filter) SDK Exceptions fangen, d.h. die Fehlerbehandlung ist weitaus besser als zu ADTF 2.x Zeiten und sollte dann zumindest eine entsprechende Meldung ausgeben, als dass das System abstürzt. Wir sind zudem weiter bestrebt noch stabiler dahingehend zu werden und bei der Fehlersuche zu unterstützen.

Um getrennte Use Cases (Visualisierung und Recording) sicher voneinander zu trennen, sollte in [#12113#note-2](#) alles gesagt sein.