

## Public Support - Support Request #10007

### Communication between dSpace VEOS and ADTF 3.x

2020-01-23 13:15 - hidden

<b>Status:</b> Closed	
<b>Priority:</b> Normal	
<b>Category:</b>	
<b>Customer:</b> AUDI	<b>Product Issue Numbers:</b>
<b>Department:</b> EF	<b>Affected Products:</b> ADTF 3.6.2
<b>Requester's Priority:</b> Normal	<b>Platform:</b>
<b>Support Level:</b> 3rd Level	<b>Topic:</b> ADTF::MessageBus
<b>Resolution:</b> Workaround Available	<b>FAQ Links:</b>
<b>Description</b>	
<b>Supportanfrage</b>	
<p>Wir betreiben aktuell eine Co-Simulation zwischen dSpace VEOS und ADTF 2.14. Für den Datenaustausch gibt es von dSpace ein Blockset, das den ADTF-Messagebus benutzt.</p> <p>Wir möchten nun für künftige Simulationen auf ADTF 3 umstellen. Leider kann uns dSpace nicht sagen, ob die Kopplung dann auch noch funktionieren wird. Die Aussage war lediglich: Wenn der Message-Bus gleich funktioniert, wird es wahrscheinlich gehen. Leider haben wir in ADTF 3 nicht die gleichen Konfigurationsmöglichkeiten wie in ADTF 2 gefunden. Daher würde uns interessieren: Gibt es den Messagebus überhaupt noch und er heißt nur anders? Wenn ja, wie wird er konfiguriert?</p> <p>Wenn nein: Wie ist es in ADTF 3 angedacht Daten mit Simulink bzw. dSpace Simulatoren auszutauschen?</p>	
<b>Lösung</b>	
<p>Die ADTF3 Message Bus Implementierung ist genau aus der Anforderung entstanden die "alte" ADTF2 dSpace Autobox Anbindung wieder einsetzen zu können. Ich kenne dSpace VEOS jetzt selbst nicht, denke aber das sollte klappen.</p> <p>Wenn es eine allgemeine UDP Verbindung von seitens dSpace gibt, solltet Ihr das auf jeden Fall auch mit unseren "Demo UDP Sender To Non-ADTF Application" und "Demo UDP Receiver From Non-ADTF Application" Filtern bedienen können: <a href="https://support.digitalwerk.net/adtf/v3/adtf_html/page_demo_non_adtf_application_udp_sender_receiver.html">https://support.digitalwerk.net/adtf/v3/adtf_html/page_demo_non_adtf_application_udp_sender_receiver.html</a></p> <p>Bzgl. ADTF ICP Sinks und Sourcen: Die Kommunikation darin werden wir aus verschiedenen Gründen nicht offenlegen, wir bieten aber eine C++Library zu Integration in nicht-ADTF Applikationen an. Diese basiert aber auf einem gekapselten ADTF3 Subsystem, sodass sie nur auf den von ADTF3 direkt unterstützen Plattformen lauffähig ist: <a href="https://support.digitalwerk.net/adtf/v3/adtf_html/page_demo_adtf_ipc_integration.html">https://support.digitalwerk.net/adtf/v3/adtf_html/page_demo_adtf_ipc_integration.html</a></p>	
<b>Related issues:</b>	
Related to Public Support - Support Request #10828: Timing behaviour in ADTF ...	<b>Closed</b>

#### History

##### #1 - 2020-01-23 14:03 - hidden

- Project changed from Public Support to 11
- Status changed from New to In Progress
- Topic set to ADTF::MessageBus
- Affected Products ADTF 3.6.2 added

##### #2 - 2020-01-23 14:44 - hidden

- Status changed from In Progress to Customer Feedback Required
- Resolution set to Workaround Available
- Customer set to AUDI
- Department set to EF

Hallo Franz,

der aus ADTF 2.x bekannte Message Bus wurde in ADTF 3.x durch IPC angepasst, eine Übersicht über die Änderungen findest du u.a. hier:

- [https://support.digitalwerk.net/adtf/v3/keytopics/changes\\_to\\_adtf2.html](https://support.digitalwerk.net/adtf/v3/keytopics/changes_to_adtf2.html)

Das macht vieles einfacher, bedeutet aber für Non-ADTF Applikationen, die mit ADTF 2.x Message Bus kommunizieren, muss eine neue Lösung geschaffen werden.

Das liegt in diesem Fall bei dSpace, dass sie das tun. Oder eine Schnittstelle bereitstellen, seitens ADTF sind uns hier die Hände gebunden.

Für diese Use Cases gibt es eine interne Open Source Beispiel Anbindung des Message Bus aus ADTF 2.x an ADTF 3.x, die ihr dazu verwenden könnt:

- [https://www.cip.audi.de/bitbucket/projects/OPENDEV/repos/adtf\\_community/browse](https://www.cip.audi.de/bitbucket/projects/OPENDEV/repos/adtf_community/browse)

Ob ihr damit hinkommt, und mit UDP auskommt, kann ich nicht sagen.

Ihr könnt aber gern daran mitarbeiten, die saubere Lösung wäre eigentlich eine direkte ADTF 3.x Anbindung.

Ich hoffe das beantwortet deine Fragen, mehr kann ich dir an dieser Stelle leider nicht out of the box anbieten.

### #3 - 2020-01-23 16:15 - hidden

Hallo Florian,

vielen Dank für die Infos und das Beispiel. Wir werden probieren, ob wir damit zurecht kommen.

Eine schnelle Lösung seitens dSpace ist hier leider auch nicht zu erwarten. Kannst du uns deshalb vielleicht ein paar mehr Informationen über IPC UDP zukommen lassen kannst. Vielleicht ist es ja mit verhältnismäßigem Aufwand implementierbar. Wenn nicht, brauchen wir langfristig ohnehin eine Lösung seitens dSpace.

Von der Verbindung her werden wir wohl erst mal bei UDP bleiben, da es für allgemeine UDP-Verbindungen definitiv ein Blockset von dSpace gibt.

Viele Grüße

Franz

### #4 - 2020-01-24 10:03 - hidden

Hi Frank,

die ADTF3 Message Bus Implementierung ist genau aus der Anforderung entstanden die "alte" ADTF2 dSpace Autobox Anbindung wieder einsetzen zu können. Ich kenne dSpace VEOS jetzt selbst nicht, denke aber das sollte klappen.

Wenn es eine allgemeine UDP Verbindung von seitens dSpace gibt, solltet Ihr das auf jeden Fall auch mit unseren "Demo UDP Sender To Non-ADTF Application" und "Demo UDP Receiver From Non-ADTF Application" Filtern bedienen können:

[https://support.digitalwerk.net/adtf/v3/adtf\\_html/page\\_demo\\_non\\_adtf\\_application\\_udp\\_sender\\_receiver.html](https://support.digitalwerk.net/adtf/v3/adtf_html/page_demo_non_adtf_application_udp_sender_receiver.html)

Bzgl. ADTF ICP Sinks und Sourcen: Die Kommunikation darin werden wir aus verschiedenen Gründen nicht offenlegen, wir bieten aber eine C++Library zu Integration in nicht-ADTF Applikationen an. Diese basiert aber auf einem gekapselten ADTF3 Subsystem, sodass sie nur auf den von ADTF3 direkt unterstützen Plattformen lauffähig ist: [https://support.digitalwerk.net/adtf/v3/adtf\\_html/page\\_demo\\_adtf\\_ipc\\_integration.html](https://support.digitalwerk.net/adtf/v3/adtf_html/page_demo_adtf_ipc_integration.html)

Grüße,

Martin

### #5 - 2020-01-27 10:00 - hidden

Hallo,

vielen Dank für die Infos. Wenn es mit der Autobox klappt, sollte es mit VEOS auch gehen. Wir werden diesen Ansatz als erstes probieren.

Dazu haben wir uns auch schon die Demo angeschaut, die ihr uns gesagt habt:

[https://www.cip.audi.de/bitbucket/projects/OPENDEV/repos/adtf\\_community/browse](https://www.cip.audi.de/bitbucket/projects/OPENDEV/repos/adtf_community/browse)

Dazu hätten wir noch eine Frage: Wir haben in dem Beispiel die Plugin-Datei nicht gefunden. Muss man sich das selbst kompilieren? Oder liegt das wo anders schon fertig ab? Wir warten leider immer noch auf unsere Visual-Studio Installation. :-/

Außerdem habe ich in der Demo jetzt nur den Fall gefunden, dass ADTF etwas empfängt. Gibt es auch ein Beispiel zum Senden von Daten?

Viele Grüße

Franz

### #6 - 2020-01-27 10:38 - hidden

Hallo Franz,

Wir haben in dem Beispiel die Plugin-Datei nicht gefunden. Muss man sich das selbst kompilieren? Oder liegt das wo anders schon fertig ab? Wir warten leider immer noch auf unsere Visual-Studio Installation.

Richtig, dass ist ein reines Git Repo, Source Code, keine Binaries.

Wir können sie euch gerne kompilieren, wenn ihr derzeit keine Build Umgebung habt.

Ich nehme an es geht um Windows ?

Außerdem habe ich in der Demo jetzt nur den Fall gefunden, dass ADTF etwas empfängt. Gibt es auch ein Beispiel zum Senden von Daten?

Es gehen beide Richtungen, im src Ordner findet ihr die beiden ADTF Streaming Services:

- Source zum Empfang der Daten
- Sink für die Senderichtung

**#7 - 2020-01-27 11:00 - hidden**

Hi,

das wäre super, wenn ihr uns die einmal kompilieren könntet!  
Ja genau, wir nutzen momentan Windows. Danke!

Ok, dann schau ich mir das nochmal genauer an.

Viele Grüße  
Franz

**#8 - 2020-01-27 15:07 - hidden**

- File *message\_bus\_compatibility\_adtf3.zip* added

Hallo Franz,

anbei die Binaries.

**#9 - 2020-01-28 14:15 - hidden**

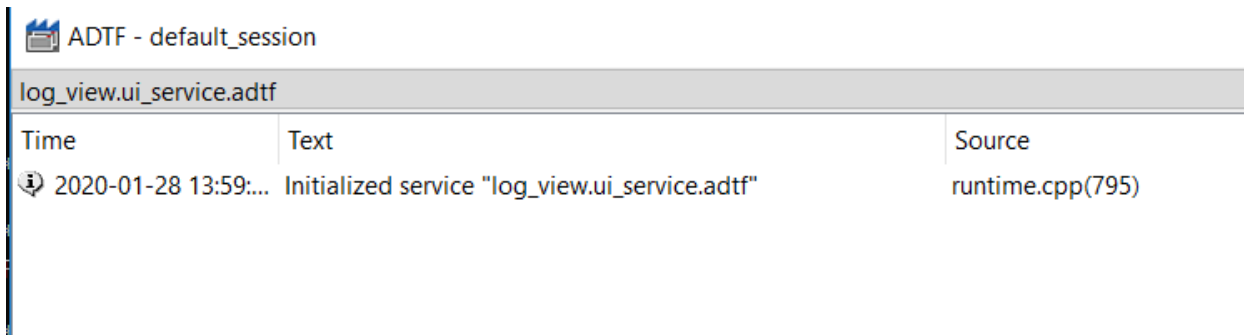
- File *image001.png* added

- File *adtf1.description* added

Hallo Florian,

nochmal danke für die Binaries. Damit funktioniert die Demo jetzt wunderbar.

Leider aber nur, wenn wir sie so belassen, wie sie ist. Wir möchten nun gerne eine von dSpace generierte description-Datei einbinden. Wenn wir aber den Dateinamen in der „default\_system.adtfproperties“ auf die neue Datei anpassen, stürzt ADTF bereits bei der Initialisierung ab. Ich habe dir einen Screenshot und das description-File angehängt.



Kann es sein, dass diese Dateien nicht mehr kompatibel sind? Oder muss man noch etwas anderes beachten?

Viele Grüße  
Franz

**#10 - 2020-01-28 16:57 - hidden**

Hi Franz,

ich kanns nicht ganz nachvollziehen, bzw. bei mir klappts. Hier mein Vorgehen:

- Ich hab das Beispiel ADTF3 Projekt geöffnet.
- Beim Media Description Service in den Properties euer .description File angegeben (anstatt des bisherigen)
- Bei der Source dann unter "struct\_name" "tmy\_adtf\_message" eingetragen und das ganze laufen gelassen.

Dann läuft es bei mir, aber mangels Gegenstelle sehe ich natürlich keine Daten.

Kannst Du euren Absturz und euer Vorgehen noch genauer beschreiben?

Grüße,

Maritn

**#11 - 2020-01-29 13:00 - hidden**

Hallo Martin,

Ok, das wundert mich. Wir werden es nochmal probieren und es dir dann genauer beschreiben, wenn es wieder nicht geht. Die Kollegin ist nur heute leider krank.

Hast du beim stream\_name auch etwas angepasst?

Viele Grüße  
Franz

**#12 - 2020-01-29 13:42 - hidden**

Hast du beim stream\_name auch etwas angepasst?

nein habe ich nicht. Das bestimmt "nur" den Namen der Messages auf dem Bus. Der sollte für jede Source/Sink eindeutig sein. Da weiß ich leider nicht genau, was die Autobox da schickt bzw was sie erwartet. Da muss man letztendlich den Namen des Ports des Subgraphen aus der entsprechenden ADTF2 Config eintragen.

Grüße,

Martin

**#13 - 2020-02-03 16:15 - hidden**

Hallo Martin,

danke für den Tipp. Damit läuft es jetzt endlich und wir können sogar auch senden. ☺

Ich hätte nun noch 3 Fragen zur best practise:

Muss man jetzt je Stream ein eigenes Decode-Plugin einbauen oder kann der auch mehrere Streams? Gilt das gleiche für die Sink?

Wie werden die Descripton-Files der Sink erstellt. In ADTF2 gab es da einen Editor. Gibt es hier auch etwas ähnliches, oder muss man sich die Dateien selbst schreiben?

Viele Grüße  
Franz

**#14 - 2020-02-04 13:24 - hidden**

Muss man jetzt je Stream ein eigenes Decode-Plugin einbauen oder kann der auch mehrere Streams? Gilt das gleiche für die Sink?

Meinst Du mit Decode Plugin die Source?

Ja da braucht's für jeden Stream eine eigene. Dadurch gibts ein 1:1 mapping zwischen den Properties und den Datenströmen. Channel können sich aber alle Sinks und Sources ein und dieselbe Instanz teilen.

Wie werden die Descripton-Files der Sink erstellt. In ADTF2 gab es da einen Editor. Gibt es hier auch etwas ähnliches, oder muss man sich die Dateien selbst schreiben?

in <adtf\_dir>/3rdparty/ddl\_editor/bin gibts wieder so einen Editor. Ihr könnt aber auch den von ADTF2 weiterverwenden.

Es gibt aber eigentlich keine Grund da neue Description Dateien zu erstellen. Die aus eurer ADTF2 Konfiguration könnt ihr unverändert übernehmen.

Grüße,

Martin

**#17 - 2020-02-13 15:13 - hidden**

- Project changed from 11 to 4

- Subject changed from Kopplung von dSpace VEOS mit ADTF 3 to Communication between dSpace VEOS and ADTF 3.x

- Description updated

- Status changed from Customer Feedback Required to To Be Closed

- Private changed from Yes to No

**#18 - 2020-03-18 11:15 - hidden**

- Related to Support Request #10828: Timing behaviour in ADTF Message Bus added

**#20 - 2020-07-07 11:12 - hidden**

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

**#22 - 2020-07-07 12:49 - hidden**

- Status changed from To Be Closed to Closed

**#23 - 2020-07-07 12:53 - hidden**

- Project changed from 4 to Public Support

## Files

---

message_bus_compatibility_adtf3.zip	14 MB	2020-01-27	hidden
image001.png	9.74 KB	2020-01-28	hidden
adtf1.description	2.74 KB	2020-01-28	hidden