

Public Support - Support Request #10608

Improve IRuntime::GetObject error messages

2020-02-26 12:15 - hidden

Status:	Closed		
Priority:	Normal		
Category:			
Customer:	ELEKTROBIT	Product Issue Numbers:	https://www.cip.audi.de/jira/browse/ACORE-10428
Department:	SUPPORT	Affected Products:	
Requester's Priority:	Normal	Platform:	
Support Level:	3rd Level	Topic:	ADTF::Common
Resolution:	Product Issue Opened	FAQ Links:	

Description

Supportanfrage

wir haben hier eine Anfrage bekommen die ich gerne an euch weitergeben möchte:

"Wenn ein ADTF System nicht alle Abhängigkeiten (in der richtigen Reihenfolge und Runlevel) bereit stellt, bekommt der Nutzer derzeit oft keine Informationen über die Ursache.

Beispiel: Der ADTF2 SessionManager benötigt den Namespace Service. Fehlt der Namespace Service, gibt es nur einen wenig hilfreichen Fehlercode.

Idee: Erweiterung von ucom::IRuntimeHook, sodass Aufrufe von ucom::IRuntime::GetObject() nachvollzogen werden können. Zu beantwortende Frage:

- Welche Instanz ruft GetObject mit welchen Parametern auf (sollte die anfragende Instanz nicht ermittelt werden können, wären die Parameter allein auch schon hilfreich)

Diese Anforderung ist aus meiner Sicht sowohl für ADTF2 als auch ADTF3 wichtig."

Viele Grüße,
Anja

Lösung

ACORE-10428 für bessere Fehlermeldung eröffnet.

Zum damit verbundenen Absturz in der Display Toolbox:

Das ist ein bekanntes Problem und wird in der Display Toolbox 3.5.0 gelöst sein:

- siehe <https://support.digitalwerk.net/projects/download-center/wiki/KnownProblems#ADTF-Display-Toolbox-Known-Problems>

Die Services werden nicht im richtigen Runlevel geladen und haben auch keine Abhängigkeit geladen.
Workaround aktuell ist die Plugin Descriptions entsprechend händisch zu fixen.

```
...
<required_interfaces>
  ...
  <interface_description>
    <iid>signal_registry.flash.services.adtf.iid</iid>
  </interface_description>
  ...
  <interface_description>
    <iid>signal_registry.ant.services.adtf.iid</iid>
  </interface_description>
  ...
</required_interfaces>
```

```
...  
<runlevel>session</runlevel>  
...
```

Alternativ kann man natürlich auch alles pro Anwendungsfall im System Editor machen, die Anpassung der Plugin Descriptions der drei UI Services ist aber sicherlich die nachhaltigere "Lösung" bis zur Display TB 3.5.0.

History

#1 - 2020-02-26 13:07 - hidden

- *Project changed from Public Support to 7*
- *Status changed from New to Customer Feedback Required*
- *Topic set to ADTF::Common*
- *Customer set to ELEKTROBIT*
- *Department set to SUPPORT*

Hallo Anja,

wie du ja sicher weißt ist das Dependency Handling eines der Features in ADTF 3.x.

Jedes Plugin bzw. Komponente spezifiziert seine Abhängigkeiten (REQUIRED_INTERFACES), diese werden in die Plugin Description auch exportiert und vom CE entsprechend ausgewertet.

Dadurch wird sichergestellt, dass Laufzeitabhängigkeiten in der korrekten Reihenfolge mit dem geforderten Runlevel im System File gesetzt werden, hier muss der Endanwender keine Hand mehr anlegen.

Im ADTF 2.x sind uns hier bekanntlich die Hände gebunden.

#2 - 2020-02-28 09:00 - hidden

- *File image001.png added*

Hallo Florian,

die Dependency Resolution im Configurationseditor ist zwar gut und auch schon besser geworden, löst aber nicht alle Probleme, z.B. werden die Run Level nicht berücksichtigt. Also selbst wenn das benutzt wurde, kann es zu Fehlern kommen. Dann gibt es natürlich noch Fehler der Benutzer und Entwickler, die auch zu sinnvollen Fehlermeldungen führen sollten.

Die Fehlermeldungen wegen nicht gefunden Interfaces sind aber sehr unspezifisch. Man erfährt lediglich, dass der GetObject Aufruf fehl schlug.

image001.png

Sieht so aus, also ob im rpc_object_registry.cpp die entsprechenden Exceptions abgefangen werden.

Wäre es möglich:

Das Handling mit den REQUIRED_INTERFACES so ausbauen, dass auch Runtime-Checks mit guten Fehlermeldungen beim laden/initialisieren des Moduls durchgeführt werden. Evtl. zu strikt, da man so nicht die Prüfung abhängig von dem Entsprechenden Initialisierungsschritt macht.

Deshalb alternativ: Die Fehlermeldung mit der angefragten iid bzw. cid ergänzen.

Dann ist da noch das Problem hierbei ist, dass das kein Automatismus ist.

Der Programmierer kann den Eintrag für REQUIRED_INTERFACES versehentlich weglassen und es dennoch benutzen.

In dem Fall fällt die Fehlersuche schwer, wenn man den Source Code nicht hat.

Grüße,
Anja

#3 - 2020-03-02 12:20 - hidden

Hallo Anja,

die Dependency Resolution im Configurationseditor ist zwar gut und auch schon besser geworden, löst aber nicht alle Probleme, z.B. werden die Run Level nicht berücksichtigt.

Also selbst wenn das benutzt wurde, kann es zu Fehlern kommen.

Das stimmt nicht, was meinst du damit ?

In jedem Service wird ein Default Runlevel definiert, das gibt der Entwickler vor und so landet es auch in der Plugin Description.

Siehe https://support.digitalwerk.net/adtf/v3/guides/sdk_generate_plugin_description.html

Wer das ändert, bricht den Vertrag und muss sich dessen bewusst sein. Oder verstehe ich dich falsch ?

Was wir noch anbieten wollen:

- Einen Hinweis und Reset Funktion, wenn das default runlevel geändert wird (ACORE-10426)
- Dass der Launcher bzgl. noch mehr checks und Ausgabe (Abhängigkeiten, default Vergleich, etc) zur Laufzeit durchführt beim Start (ACORE-10425)

Dann gibt es natürlich noch Fehler der Benutzer und Entwickler, die auch zu sinnvollen Fehlermeldungen führen sollten. Die Fehlermeldungen wegen nicht gefunden Interfaces sind aber sehr unspezifisch. Man erfährt lediglich, dass der GetObject Aufruf fehl schlug.

image001.png

Sieht so aus, also ob im `rpc_object_registry.cpp` die entsprechenden Exceptions abgefangen werden.

Das hat aber imho nichts mit Dependencies zu tun, hier wird versucht sich an ein System zu docken (Control, GUI Control), dass nicht existiert, richtig ?

Also wenn du nur mit Launcher ausführst, hättest du im Log die eigentliche Fehlermeldung davor und diese eben nicht ?

@Martin: Können wir hier besser werden ? Die leere Angabe ist sicherlich nichtssagend...

Wäre es möglich:

Das Handling mit den `REQUIRED_INTERFACES` so ausbauen, dass auch Runtime-Checks mit guten Fehlermeldungen beim laden/initialisieren des Moduls durchgeführt werden. Evtl. zu strikt, da man so nicht die Prüfung abhängig von dem Entsprechenden Initialisierungsschritt macht.

Das soll wie gesagt in ACORE-10425 noch tiefer gehen

Deshalb alternativ: Die Fehlermeldung mit der angefragten iid bzw. cid ergänzen.

Wie gesagt, ich denke das Problem hier ist dass gar kein ADTF System besteht, das per RPC angesprochen wird

Dann ist da noch das Problem hierbei ist, dass das kein Automatismus ist.

Der Programmierer kann den Eintrag für `REQUIRED_INTERFACES` versehentlich weglassen und es dennoch benutzen.

In dem Fall fällt die Fehlersuche schwer, wenn man den Source Code nicht hat.

Das hat man an so vielen Stellen in der Programmierung, das gibt aber C++ ohne Reflection leider nicht her und wir wollen nicht den Code parsen. Das sind Makros die einfach verwendet werden müssen im ADTF Kontext (analog zu Header includes), gilt ja auch für `ADTF_PLUGIN`. Wir haben versucht, dass hier zu dokumentieren:

- https://support.digitalwerk.net/adtf/v3/adtf_html/page_versioned_namespaces.html

#4 - 2020-03-03 10:26 - hidden

- Resolution set to Product Issue Opened

- Product Issue Numbers set to <https://www.cip.audi.de/jira/browse/ACORE-10428>

Hallo Anja,

ich habe für aussagekräftigere Fehlermeldungen das Ticket <https://www.cip.audi.de/jira/browse/ACORE-10428> erstellt. Dadurch sollte man jetzt immer zumindest die ID des gesuchten Interfaces in der Fehlermeldung bekommen (und auch die Object IDs der abgefragten Objekte).

Grüße,

Martin

#5 - 2020-03-04 08:28 - hidden

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

Hallo Anja,

Martin wrote:

ich habe für aussagekräftigere Fehlermeldungen das Ticket <https://www.cip.audi.de/jira/browse/ACORE-10428> erstellt. Dadurch sollte man jetzt immer zumindest die ID des gesuchten Interfaces in der Fehlermeldung bekommen (und auch die Object IDs der abgefragten Objekte).

nach Umsetzung von ACORE-10428 sieht die Meldung nun so aus:

ADTF Runtime Exception

Exception occurred while running adtf:

Result code '-20' (ERR_NOT_FOUND) - default_streaming_graph.Demo Virtual Clock Input: No object with the requested interface found:

'logging.system.adtf': Object does not implement interface 'reference_clock.giant.streaming.adtf.iid'.

'license_manager.system.adtf': Object does not implement interface 'reference_clock.giant.streaming.adtf.iid'.

'macro_resolver.system.adtf': Object does not implement interface 'reference_clock.giant.streaming.adtf.iid'.

```
'rpc_object_server_registry.system.adtf': Object does not implement interface 'reference_clock.giant.streaming.adtf.iid'.
'session_manager.system.adtf': Object does not implement interface 'reference_clock.giant.streaming.adtf.iid'.
'ADTF Kernel Service': Object does not implement interface 'reference_clock.giant.streaming.adtf.iid'.
[File: c:\dev\acore\source\src\libraries\ucom3\src\runtime.cpp] [Line: 1244] [Func: adtf::ucom::catwo::detail:
:cRuntime::cRuntimePrivate::GetObject]
[File: c:\dev\acore\source\src\examples\src\adtf\streaming_services\sources\virtual_clock\demo_virtual_clock.c
pp] [Line: 62] [Func: cDemoVirtualClock::StartStreaming]
[File: c:\dev\acore\source\src\libraries\streaming3\src\streaming_graph.cpp] [Line: 387] [Func: adtf::streamin
g::catwo::cStreamingGraph::SetStateImpl]
[File: c:\dev\acore\source\src\plugins\adtf_core\session_manager.cpp] [Line: 75] [Func: set_session_streaming_
state]
[File: c:\dev\acore\source\src\plugins\adtf_core\session_manager.cpp] [Line: 169] [Func: cSessionManager::Runt
imeHook]
```

Und sollte in diesem Fall Aufschluss geben, dass kein Objekt (Service) geladen ist, der die Version der Reference Clock bereitstellt, welche von Demo Virtual Clock angefragt wird.

#7 - 2020-03-04 09:08 - hidden

Hallo Florian,

ich kommentiere der Einfachheit halber:

Florian Roth wrote:

Hallo Anja,

die Dependency Resolution im Configurationseditor ist zwar gut und auch schon besser geworden, löst aber nicht alle Probleme, z.B. werden die Run Level nicht berücksichtigt.
Also selbst wenn das benutzt wurde, kann es zu Fehlern kommen.

Das stimmt nicht, was meinst du damit ?

"

Natürlich kommt die Fehlermeldung von einem gelaunchten System, woher denn sonst. Ein nicht gelaunchtes System kann keine Fehlermeldungen ausgeben.

Das Beispiel kam aus den Device Toolbox Plugins. Dort wird immerhin eine Exception geschmissen, wenn das required interface nicht gefunden wurde.

Schlimmer ist es in der Display Toolbox:

Er soll doch mal in eins der ADTF Example-Projekten die Signal Tree View Service hinzufügen. Da gibt's nicht mal ne Exception. Lediglich die Info dass der Service nicht geladen werden konnte und ein Absturz.

Es klappt erst, wenn man der Signal Registry Service im Runlevel 1 lädt.

"

Grüße,
Anja

#8 - 2020-03-04 09:51 - hidden

Hallo Anja,

irgendwie passt dein Text/Antwort nicht zum zitierten Text...
Ich versuche trotzdem darauf einzugehen:

Natürlich kommt die Fehlermeldung von einem gelaunchten System, woher denn sonst. Ein nicht gelaunchtes System kann keine Fehlermeldungen ausgeben.

Ich habe mich falsch ausgedrückt und habe die Fehlermeldung falsch interpretiert, ich kenne eine ähnliche bzgl. RPC in Zusammenhang mit ADTF (GUI) Control und bitte vielmals um Entschuldigung.

Das sollte sich doch aber im weiteren Ticketverlauf auch mit den Ausführungen von Martin schon korrigiert haben, ebenso mit der Umsetzung von ACORE-10428 gelöst sein, siehe nochmal [#10608#note-5](#).

So ist es nun im master und landet auch schon in der 3.7.

Das Beispiel kam aus den Device Toolbox Plugins. Dort wird immerhin eine Exception geschmissen, wenn das required interface nicht gefunden wurde.

Sollte nun wie gesagt mit ACORE-10428 aufschlussreicher sein.

Schlimmer ist es in der Display Toolbox:

Er soll doch mal in eins der ADTF Example-Projekten die Signal Tree View Service hinzufügen. Da gibt's nicht mal ne Exception. Lediglich die Info dass der Service nicht geladen werden konnte und ein Absturz.

Es klappt erst, wenn man der Signal Registry Service im Runlevel 1 lädt.

Das ist ein bekanntes Problem und wird in der Display Toolbox 3.5.0 gelöst sein:

- siehe <https://support.digitalwerk.net/projects/download-center/wiki/KnownProblems#ADTF-Display-Toolbox-Known-Problems>

Die Services werden nicht im richtigen Runlevel geladen und haben auch keine Abhängigkeit geladen. Workaround aktuell ist die Plugin Descriptions entsprechend händisch zu fixen.

```
...
<required_interfaces>
...
<interface_description>
  <iid>signal_registry.flash.services.adtf.iid</iid>
</interface_description>
...
<interface_description>
  <iid>signal_registry.ant.services.adtf.iid</iid>
</interface_description>
...
</required_interfaces>
...
<runlevel>session</runlevel>
...
```

Alternativ kann man natürlich auch alles pro Anwendungsfall im System Editor machen, die Anpassung der Plugin Descriptions der drei UI Services ist aber sicherlich die nachhaltigere "Lösung" bis zur Display TB 3.5.0.

Allgemein bitte ich darum den Ton der mir/uns hier entgegengebracht wird zu hinterfragen. Das gebührt schon allein der Respekt, wir versuchen hier nach besten Wissen und Gewissen zu helfen. Ebenso bitte ich auf die Rückfragen einzugehen oder eine Rückmeldung zu geben, wann das geschieht oder obsolet ist, anstatt neue Baustellen (Stichwort Toolboxes) aufzumachen. Das darfst du gerne so weitergeben und ich appelliere an unsere partnerschaftliche Zusammenarbeit.

Passt das nun mit der Fehlermeldung/Umsetzung oder nicht ?

Desweiteren ist noch folgender Punkt offen:

Anja Winkler wrote:

die Dependency Resolution im Configurationseditor ist zwar gut und auch schon besser geworden, löst aber nicht alle Probleme, z.B. werden die Run Level nicht berücksichtigt. Also selbst wenn das benutzt wurde, kann es zu Fehlern kommen.

Florian Roth wrote:

Das stimmt nicht, was meinst du damit ?
In jedem Service wird ein Default Runlevel definiert, das gibt der Entwickler vor und so landet es auch in der Plugin Description. Siehe https://support.digitalwerk.net/adtf/v3/guides/sdk_generate_plugin_description.html
Wer das ändert, bricht den Vertrag und muss sich dessen bewusst sein. Oder verstehe ich dich falsch ?

#9 - 2020-03-04 13:04 - hidden

Hallo Florian,

also vielen Dank für deine Antwort. Es tut mir leid das auf Grund meiner Auslastung ich nicht alle Kommunikation überarbeitet habe oder sie auch durcheinander geraten ist. Natürlich arbeiten wir hier partnerschaftlich zusammen, das steht hoffentlich außer Frage. Ich werde die Anfragen wieder gewissenhafter formulieren.

Martins Vorschläge passen soweit für uns, du warst schneller mit deiner Antwort als ich die Rückmeldung hatte.

Zu dem weiteren offenen Punkt. Ich denke das ist hier ein Missverständnis das aus der Geschichte mit den Toolboxes resultierte und sich nun ja aufgelöst hat, bzw. auf Grund von Martins und deiner weitem Antwort gelöst hat.

Vielen Dank für deine/eure Hilfe,
Anja

#10 - 2020-03-04 13:35 - hidden

- Project changed from 7 to Public Support
- Subject changed from EBPRODUCTSUPPORT-6326 [EB Germany]Informationen über _runtime->GetObject()-Aufrufe sollten nachvollziehbar sein to Improve IRuntime::GetObject error messages
- Description updated
- Status changed from Customer Feedback Required to To Be Closed
- Private changed from Yes to No

Hallo Anja,

Natürlich arbeiten wir hier partnerschaftlich zusammen, das steht hoffentlich außer Frage.

Auf jeden Fall

Martins Vorschläge passen soweit für uns, du warst schneller mit deiner Antwort als ich die Rückmeldung hatte.
Zu dem weiteren offenen Punkt. Ich denke das ist hier ein Missverständnis das aus der Geschichte mit den Toolboxes resultierte und sich nun ja aufgelöst hat, bzw. auf Grund von Martins und deiner weitem Antwort gelöst hat.

Gut, dann schließe ich das Ticket damit ab

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

- Status changed from To Be Closed to Closed

Files

image001.png	56.3 KB	2020-02-28	hidden
--------------	---------	------------	--------