

Public Support - Support Request #6936

adtf::util namespace is missing in the API documentation

2019-04-23 16:00 - hidden

Status:	Closed		
Priority:	Normal		
Category:			
Customer:	VW	Product Issue Numbers:	https://www.cip.audi.de/jira/browse/ACORE-9994
Department:	IAV	Affected Products:	ADTF 3.5.0
Requester's Priority:	Normal	Platform:	
Support Level:	3rd Level	Topic:	ADTF::Doc
Resolution:	Product Issue Opened	FAQ Links:	

Description

Supportanfrage

Today i tried to find

```
adtf::util::cString
```

in the documentation at: https://support.digitalwerk.net/adtf/v3/adtf_html/index.html
But the whole adtf::util:: is not visible there.

Lösung

Product issue ACORE-9994

Workaround

Use std::string

```
std::string strMyValue;  
GetIStringFunctionThatYouWantToUse(adtf_string_intf(strMyValue));
```

History

#1 - 2019-04-23 16:17 - hidden

- Project changed from Public Support to 20

- Subject changed from adtf::util namespacce is missing in the API documentation to adtf::util namespace is missing in the API documentation

- Topic set to ADTF::Doc

#2 - 2019-04-23 17:36 - hidden

- Status changed from New to Customer Feedback Required

Hallo Sebastian,

I guess we can proceed in German ? If not, let me know...

Das hat build Gründe.

Die Utils sind nun ein eigenständiges Paket, deshalb wird die Doku nicht mit generiert, sondern nur das Paket geliefert.

Du musst dich also auf die Header an sich berufen -> <ADTF_DIR>/pkg/a_utils/include

Aber generiert wird hier nichts mehr, wir wollen eigentlich eher utils unabhängig arbeiten...

Was genau fehlt dir ?

Was hättest du wissen wollen ?

#3 - 2019-04-24 08:15 - hidden

Hallo,

ich wollte eine Funktion benutzen, die eine rvalue referenz auf IString liefert. Um einen String damit zu füllen brauchte ich natürlich erst mal ein konkretes Objekt.

Erst nach längerer Suche in den Headern konnte ich ein Beispiel finden, in Form eines `adtf::util::cString`, weil in der API-Dokumentation nur Klassen verlinkt waren wie `adtf_string< T >` die nicht so einfach zu benutzen sind.

Heute habe ich jetzt auch noch ein Beispiel in der Dokumentation gefunden:

https://support.digitalwerk.net/adtf/v3/adtf_html/string_intf_8h.html#a8be13bc6e1ac810e9b27b74c27eece0c

Ich denke es wäre sehr hilfreich, wenn das in der description zu IString verlinkt werden könnte. Oder soll man die Utils nicht mehr benutzen. Ich verstehe den Sinn noch nicht so ganz ;-)

In welchem package befinden sich denn die Utils genau? Wenn ich zu

https://support.digitalwerk.net/adtf/v3/adtf_html/page_delivered_sdk_packages.html gehe, sehe ich da nichts mit Utils.

Grüße
Sebastian Kratzert

#4 - 2019-04-24 08:25 - hidden

- Resolution set to Product Issue Opened

- Product Issue Numbers set to <https://www.cip.audi.de/jira/browse/ACORE-9994>

Hallo Sebastian,

In welchem package befinden sich denn die Utils genau? Wenn ich zu

https://support.digitalwerk.net/adtf/v3/adtf_html/page_delivered_sdk_packages.html gehe, sehe ich da nichts mit Utils.

Wie gesagt, für die Utils wird derzeit keine Doxygen/html aus den Headern generiert, da es ein eigenständige Package ist und "nur" mitgeliefert wird. Deshalb gehen auch keine Verweise.

Die Header sind aber trotzdem da, wie bereits verwiesen allerdings "nur" in der Lieferung -> `<ADTF_DIR>/pkg/a_utils/include`
Aktuell müsstest du dich damit abfinden bzw. selbst generieren mit doxygen.

Ich denke es wäre sehr hilfreich, wenn das in der description zu IString verlinkt werden könnte.

Ich eine Ticket (ACORE-9994) erstellt, in dem eine doxygen Doku für die utils evaluiert werden soll.
Das muss geprüft werden.

Oder soll man die Utils nicht mehr benutzen

Das habe ich so nicht gesagt, nur darauf verwiesen, dass wir davon weg wollen.

Es gibt eigentlich nichts was man nicht auch mit Standard Library lösen könnte, z.B. `std::string`.

Die Utils sind eigentlich ein ADTF 2.x Gebilde bzgl. Binärkompatibilität und deshalb auch in ADTF 3.x.

Das ist aber nur intern über Interface Grenzen hinweg relevant.

Aber in ADTF 3.x werden wir sie natürlich nicht mehr los, demnach sind sie drin und du kannst sie benutzen, ich würde aber nicht ausschließen, dass "unwichtige" Legacy, deprecated oder wegfallen werden.

#5 - 2019-04-25 13:15 - hidden

Es gibt eigentlich nichts was man nicht auch mit Standard Library lösen könnte, z.B. `std::string`.

Ich habe gerade kein ADTF3 zur Hand... Wäre es dann also möglich vorher von mir genannte Beispiel mit `std::string` zu implementieren?
Dann wären die utils tatsächlich für mich auch nicht nötig.

Also so:

```
struct foo
{
    tResult GetString(IString&& strValue)
    {
        return strValue.Set("bar");
    }
};

std::string strMyValue = "foo";
foo oFooInst;
oFooInst.GetString(adtf_string_intf(strMyValue));
```

Das Ticket kann von mir aus dann geschlossen werden...

Vielen Dank!
Sebastian Kratzert

#6 - 2019-04-25 13:53 - hidden

Hallo Sebastian,

OK ? Wenn du kein ADTF zur Hand hast, wie willst du dann entwickeln ?
Nun gut.

Wenn ich deinen Use Case insgesamt richtig verstehe, kannst du das mit einem Zweizeiler erledigen:

```
std::string strMyValue;  
GetIStringFunctionThatYouWantToUse(adtf_string_intf(strMyValue));
```

#7 - 2019-04-25 14:30 - hidden

Hallo Florian,

zurzeit entwickelt ein Student für mich. Daher hatte ich es noch nicht installiert.
Deine Antwort bzgl. Std::string stellt mich voll zufrieden.

Zu solchen Dingen würde ich mir mehr Dokumentation über das „große ganze“ wünschen. Mit der 3.5.0 und den neuen Beispielen ist es ja schon viel besser geworden.

Aber z.B. properties werden für meinen Geschmack noch zu wenig erklärt. Ich finde finde dort keinen Hinweis, ob sie auch zur Laufzeit ihren Wert ändern können und wie man sich auch change-Events registrieren kann, wie in ADTF2. Aber das wäre eher Stoff für ein neues Ticket...

#8 - 2019-04-25 15:24 - hidden

- *Project changed from 20 to Public Support*
- *Description updated*
- *Status changed from Customer Feedback Required to To Be Closed*
- *Private changed from Yes to No*

Hallo Sebastian,

zurzeit entwickelt ein Student für mich. Daher hatte ich es noch nicht installiert.

Alles klar.

Deine Antwort bzgl. Std::string stellt mich voll zufrieden.

Sehr gut !

Zu solchen Dingen würde ich mir mehr Dokumentation über das „große ganze“ wünschen. Mit der 3.5.0 und den neuen Beispielen ist es ja schon viel besser geworden.

Da bin ich bei dir und da haben (und werden wir weiterhin) viel Zeit reinstecken, ist halt Stückwerk.
Die Basissachen sollten nun ganz gut abgedeckt sein mit dem neuen Filter SDK, nun liegt es an den Guides hauptsächlich, Use Cases zu decken.

Freut uns zu hören, dass das zumindest ankommt und hilft.

Aber z.B. properties werden für meinen Geschmack noch zu wenig erklärt.

Hier passiert in der 3.6 nun einiges, u.a. das Property Handling wird aufgeräumt.
Was aktuell noch fehlt, ist dazu die Doku, hier haben wir ein Ticket in Arbeit, dass v.a. mit Hinblick auf ADTF 2.x die Use Cases abdecken soll in einem Guide (ACORE-9725).

Ich finde finde dort keinen Hinweis, ob sie auch zur Laufzeit ihren Wert ändern können und wie man sich auch change-Events registrieren kann, wie in ADTF2.

Die Property Schnittstelle ist eigentlich komplett entkoppelt worden, du kannst Properties zur Konfigurationzeit (siehe [ADTF Configuration Editor / ADTF Config Tool](#)) und auch Laufzeit (z.B. mittels [ADTF Control](#)) jederzeit ändern, der Filter muss hier eben drauf reagieren. Ein Callback a la *PropertyChanged* aus ADTF 2.x ist in ADTF 3.x eigentlich nicht mehr möglich, da der aktuelle Wert in der registrierten Property Variable verfügbar ist,

die automatisch einen Observer anmeldet. Sprich du fragst einfach den Ist-Stand ab und führst entsprechend deinen Code aus (oder auch nicht). Das passt zum Konzept und Architektur des entkoppelten Laufzeitverhaltens.

Aber du hast Recht, das wäre ein eigenes Ticket, deshalb nur nebenbei.

Allgemein sind wir immer über Anregungen zur Dokuverbesserung und Use Cases offen !!

#9 - 2020-07-07 11:08 - hidden

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

#10 - 2020-07-07 12:44 - hidden

- *Status changed from To Be Closed to Closed*