

Public Support - Support Request #17471

Prevent updating label names

2022-05-11 14:49 - hidden

Status:	Closed	Product Issue Numbers: Affected Products: Platform: Topic: ADTF::Common FAQ Links:
Priority:	Normal	
Category:		
Customer:	ELEKTROBIT	
Department:	SUPPORT	
Requester's Priority:	Normal	
Support Level:	2nd Level	
Resolution:	Solved Issue	

Description

Supportanfrage

please think twice when in future you update the names and CIDs of components since it may need a lot of followup work on updating documentation (not only for you).

Just one example from ADTF 3.14: you newly introduced the Qt5 prefix for many components. Your documentation was partially updated but you have still mixes of old and new names. Here just some example https://support.digitalwerk.net/adtf/v3/guides/tutorial_filter_javascript.html where you for example can find 2 different old variants for the brand new "Qt5 Media Description Display" in one document.

I would suggest to not change names every two or so minor versions. While this affects only documentation material it is even worse when cids get changed (like recently with some cids in the device toolbox)which makes it necessary to reconfigure the projects.

Lösung

Hi Jochen,

I was browsing a little bit further in the guide and here https://support.digitalwerk.net/adtf/v3/guides/sdk_properties_filter.html there is also the "Demo Media Description Display" in the video and "Qt Media Description Display" in the text. So I somehow would assume further locations that need to be updated.

Correct, as mentioned here -> [#17471#note-2](#)

You see my preference for a naming that reduces the update effort (why not just "Media Description Display"?). Hm ... in the toolboxes there are more Qt components ... which do not have Qt in the name yet.

This will be changed in upcoming toolbox as well.

Concerning importance: I still do not get it why for the users it is important to see in the component view, with which Qt version a component was created. (Hm maybe if one wants to support usage of different Qt major versions at the same time, but I guess we are not going there). And then when the projects were created with the Qt 5 version default names of the components and used with the Qt 6 update there is finally a confusing mismatch (so in the graph editor the component is called Qt 5 Media Description Display, but what is actually launched is the Qt 6 Media Description Display). Qt version information is needed much earlier as release information to find out which toolbox versions can be used with which ADTF version. Hm I have no final idea for this yet ... maybe a good approach would be to provide a compatibility matrix for the users: which toolbox versions are recommended/tested/should work for which ADTF versions. (Or recommended bundles).

Sorry, but you mix up to many different things and unfortunately do not understand our approach.
The **current** Qt related components are build with Qt5 and will always be !
But the plan in future is, to also support Qt6 but this not possible due to binary compatibility.
So if we support Qt6, there will be an additional Qt6 UI SDK, additional Qt6 XSystem and additonal Qt6 components besides the current Qt5 content.
To differ the content in component view and graph, we work with the label, it wont be possible create a graph for both worlds (nice

idea but we will focus a different way which will always work and is transparent for the user and dependencies). So if you drop a Qt5 Filter, it will add the Qt5 XSystem, if you add a Qt6 Filter, it will add the Qt6 XSystem. Both cannot be handled in one Session and will fail and declined by CE. Mixing Qt versions can only work in different instances / distributed system. So this is our plan and to prevent in future the naming, we start tagging with Qt5 now. If I could, I would also edit the CID but this I won't do to avoid breaking changes. So in future we will have a qt_xsystem.ui_service.adtf.cid (unfortunately not qt5_xsystem.ui_service.adtf.cid) and a qt6_xsystem.ui_service.adtf.cid which cannot be combined. That's it, hope its more clear now.

And yes, we are planning to create baselines for ADTF bundles. But right now, the tested ADTF version is the one from the build, so transparent for the user. Recommended will be always latest. Second approach for this will be our store, with the ability to only get a reduced package (codename adtf_nano) and download plugins from store. This hopefully can also help customers with version freezes to get updates and - hopefully in future as well - automatic downloads by just sharing sessions.

Concerning ARXML toolbox example project: actually beyond the topic of the ticket but when you look at it anyway, we found the following further issues. 1. CAN decoder example. Signal name wrong in property of the CAN coder filter. 2. Flexray example has no fibex nor arxml, so would not work. 3. SOMEIP example would be nice.

Yes, we adapt that, this has been already reported from your colleague (#17454)

From training side, I still think it would be not a good approach that people have to learn different names every now and then ... (ok guess it does not go so far yet that user of different ADTF 3 minor versions do not understand each other).

[...]

yeah guess we have a different approach with terminology. Updating terminology is a more involved process here with many different people and company internal terminology database involved. So for ADTF 2 (from version 2.9) I do not remember any significant naming update.

So please let me sum up to be clear:

- We won't change any class ids or interface ids in future, neither in ADTF or in toolboxes, we are fine now and have a process update to prevent this in future
- We are not planning to adapt label names in adtf core (besides components may get deprecated or succeed BETA state)
- We will adapt the toolboxes label names for the new profiles (to changes from adtf 3.14 as well)
 - Calibration TB
 - Signal Requester Filter -> XCP Signal Requester
 - XCP Decode Filter -> XCP Config Decoder Filter
 - XCP Emulator Filter -> XCP Emulator
 - XCP Encode Filter -> XCP Config Encoder Filter
 - XCP Master Filter -> XCP Master
 - XCP Service -> XCP Support Service
 - Device TB
 - (Deprecated) CAN Trace View -> (Deprecated) Qt5 CAN Trace View UI Service
 - CAN FD Trace View -> Qt5 CAN FD Trace View UI Service
 - DevTB2 Support Service -> ADTF 2.x Bus Data Deserializer Support Service
 - FlexRay Trace View -> Qt5 FlexRay Trace View UI Service
 - SOME IP Trace View -> Qt5 SOME IP Trace View UI Service
 - Display TB
 - 2D OpenGL Display -> Qt5 2D OpenGL Display
 - 3D OpenSceneGraph Display -> Qt5 3D OpenSceneGraph Display
 - Demo Signal Provider UI Service -> Demo Qt5 Signal Provider UI Service
 - Signal Scope View UI Service -> Qt5 Signal Scope View UI Service
 - Signal Table View UI Service -> Qt5 Signal Table View UI Service
 - Signal Tree View UI Service -> Qt5 Signal Tree View UI Service

History

#1 - 2022-05-12 06:47 - hidden

- Status changed from New to Customer Feedback Required

- Department set to SUPPORT

Hi Jochen,

regarding the label names, we will proceed changing them in future if it's important for upcoming plans (like here Qt5 vs Qt6) or help the customer (better spelling) or to harmonize spelling.

Of course we have to adapt them everywhere, I will recheck but as far after a quick search you found the only remaining place where its missing... If this would be the only problem in ADTF, I would have a good sleep :-)

In doxygen its easier because we can reference names using pages, in the guides its different.

This will be improved by upcoming concepts to combine the documentations in some way, so thanks for reporting.

About changing class ids we are totally in the same page, we prevent changing them.

We changed some two times (ADTF 3.4 and Device TB 3.8) so not every two minors...

Of course this should not happen and as you see we wont change class ids in the same manner than label names (see MD Display... sorry Qt5 Media Description Display vs demo_md_...) because otherwise, as you said, sessions getting incompatible.

We used the early chance to correct the substream spelling due to license issues and violations.

We are not planning to change class ids in future.

#2 - 2022-05-12 11:05 - hidden

Now I think i found out each remaining former Qt related string and replace for an upcoming delivery!

#3 - 2022-05-12 13:50 - hidden

Hi Florian,

concerning the CIDs: please consider the device toolbox updates also in the ARXML Toolbox example project. (In the one delivered to Wolfgang the old CIDs of CAN substream ... filter were used).

Concerning renaming of components. I would appreciate if one could also be more conservative here. We have marketing documents and training documents that mention certain filter and service names. Then similar things could apply in project documentation. And then these documents should not only fit to the users of a specific version of ADTF or a toolbox but for some recent range of versions.

What would be a decent concept for naming a specific filter/service in such a document without making it version specific?

I do not fully understand why renaming Qt ... to Qt5 ... or Qt6 ... is important in any way.

BR,
Jochen

#4 - 2022-05-12 14:19 - hidden

Hi Jochen,

concerning the CIDs: please consider the device toolbox updates also in the ARXML Toolbox example project. (In the one delivered to Wolfgang the old CIDs of CAN substream ... filter were used).

OK, I will have a look at it, thanks!

Concerning renaming of components. I would appreciate if one could also be more conservative here. We have marketing documents and training documents that mention certain filter and service names. Then similar things could apply in project documentation. And then these documents should not only fit to the users of a specific version of ADTF or a toolbox but for some recent range of versions.
What would be a decent concept for naming a specific filter/service in such a document without making it version specific?

To be fair, this is a different approach to our business, our documentation and training is always version specific.

But for now, I will have this in my mind and will have an eye on it before changing.

For now, the label names seem good to me and I don't see any changes.

I do not fully understand why renaming Qt ... to Qt5 ... or Qt6 ... is important in any way.

The issue was, we have a mix of with and without Qt Prefix and other components with or without ADTF prefix.

This was important to harmonize that and we use it, to use Qt5 instead of Qt for all.

Because when we create a Qt6 dependent XSystem and components, then the user can differ (as mentioned, cannot be upgraded from qt5 to qt6 without breaking binary compatibility).

But this is future.

#5 - 2022-05-17 15:51 - hidden

Hi Florian,

yeah guess we have a different approach with terminology. Updating terminology is a more involved process here with many different people and company internal terminology database involved. So for ADTF 2 (from version 2.9) I do not remember any significant naming update.

I was browsing a little bit further in the guide and here https://support.digitalwerk.net/adtf/v3/guides/sdk_properties_filter.html there is also the "Demo Media Description Display" in the video and "Qt Media Description Display" in the text. So I somehow would assume further locations that need to be updated.

From training side, I still think it would be not a good approach that people have to learn different names every now and then ... (ok guess it does not go so far yet that user of different ADTF 3 minor versions do not understand each other).

You see my preference for a naming that reduces the update effort (why not just "Media Description Display"?). Hm ... in the toolboxes there are more Qt components ... which do not have Qt in the name yet.

Concerning importance: I still do not get it why for the users it is important to see in the component view, with which Qt version a component was created. (Hm maybe if one wants to support usage of different Qt major versions at the same time, but I guess we are not going there). And then when the projects were created with the Qt 5 version default names of the components and used with the Qt 6 update there is finally a confusing mismatch (so in the graph editor the component is called Qt 5 Media Description Display, but what is actually launched is the Qt 6 Media Description Display). Qt version information is needed much earlier as release information to find out which toolbox versions can be used with which ADTF version. Hm I have no final idea for this yet ... maybe a good approach would be to provide a compatibility matrix for the users: which toolbox versions are recommended/tested/should work for which ADTF versions. (Or recommended bundles).

Concerning ARXML toolbox example project: actually beyond the topic of the ticket but when you look at it anyway, we found the following further issues. 1. CAN decoder example. Signal name wrong in property of the CAN coder filter. 2. Flexray example has no fibex nor arxml, so would not work. 3. SomeIP example would be nice.

#6 - 2022-05-18 10:04 - hidden

Hi Jochen,

I was browsing a little bit further in the guide and here https://support.digitalwerk.net/adtf/v3/guides/sdk_properties_filter.html there is also the "Demo Media Description Display" in the video and "Qt Media Description Display" in the text. So I somehow would assume further locations that need to be updated.

Correct, as mentioned here -> [#17471#note-2](#)

You see my preference for a naming that reduces the update effort (why not just "Media Description Display"?). Hm ... in the toolboxes there are more Qt components ... which do not have Qt in the name yet.

This will be changed in upcoming toolbox as well.

Concerning importance: I still do not get it why for the users it is important to see in the component view, with which Qt version a component was created. (Hm maybe if one wants to support usage of different Qt major versions at the same time, but I guess we are not going there). And then when the projects were created with the Qt 5 version default names of the components and used with the Qt 6 update there is finally a confusing mismatch (so in the graph editor the component is called Qt 5 Media Description Display, but what is actually launched is the Qt 6 Media Description Display). Qt version information is needed much earlier as release information to find out which toolbox versions can be used with which ADTF version. Hm I have no final idea for this yet ... maybe a good approach would be to provide a compatibility matrix for the users: which toolbox versions are recommended/tested/should work for which ADTF versions. (Or recommended bundles).

Sorry, but you mix up to many different things and unfortunately do not understand our approach.

The **current** Qt related components are build with Qt5 and will always be !

But the plan in future is, to also support Qt6 but this not possible due to binary compatibility.

So if we support Qt6, there will be an additional Qt6 UI SDK, additional Qt6 XSystem and additional Qt6 components besides the current Qt5 content.

To differ the content in component view and graph, we work with the label, it wont be possible create a graph for both worlds (nice idea but we will focus a different way which will always work and is transparent for the user and dependencies). So if you drop a Qt5 Filter, it will add the Qt5 XSystem, if you add a Qt6 Filter, it will add the Qt6 XSystem.

Both cannot be handled in one Session and will fail and declined by CE.

Mixing Qt versions can only work in different instances / distributed system.

So this is our plan and to prevent in future the naming, we start tagging with Qt5 now.

If I could, I would also edit the CID but this I won't do to avoid breaking changes.

So in future we will have a qt_xsystem.ui_service.adtf.cid (unfortunately not qt5_xsystem.ui_service.adtf.cid) and a qt6_xsystem.ui_service.adtf.cid which cannot be combined.

That's it, hope its more clear now.

And yes, we are planning to create baselines for ADTF bundles. But right now, the tested ADTF version is the one from the build, so transparent for the user. Recommended will be always latest. Second approach for this will be our store, with the ability to only get a reduced packae (codename adtf_nano) and download plugins from store. This hopefully can also help customers with version freezes to get updates and - hopefully in future as well - automatic downloads by just sharing sessions.

Concerning ARXML toolbox example project: actually beyond the topic of the ticket but when you look at it anyway, we found the following further issues. 1. CAN decoder example. Signal name wrong in property of the CAN coder filter. 2. Flexray example has no fibex nor arxml, so would not work. 3. SomeIP example would be nice.

Yes, we adapt that, this has been already reported from your colleague (#17454)

From training side, I still think it would be not a good approach that people have to learn different names every now and then ... (ok guess it does not go so far yet that user of different ADTF 3 minor versions do not understand each other).

[...]

yeah guess we have a different approach with terminology. Updating terminology is a more involved process here with many different people and company internal terminology database involved. So for ADTF 2 (from version 2.9) I do not remember any significant naming update.

So please let me sum up to be clear:

- We won't change any class ids or interface ids in future, neither in ADTF or in toolboxes, we are fine now and have a process update to prevent this in future
- We are not planning to adapt label names in adtf core (besides components may get deprecated or succeed BETA state)
- We will adapt the toolboxes label names for the new profiles (to changes from adtf 3.14 as well)
 - Calibration TB
 - Signal Requester Filter -> XCP Signal Requester
 - XCP Decode Filter -> XCP Config Decoder Filter
 - XCP Emulator Filter -> XCP Emulator
 - XCP Encode Filter -> XCP Config Encoder Filter
 - XCP Master Filter -> XCP Master
 - XCP Service -> XCP Support Service
 - Device TB
 - (Deprecated) CAN Trace View -> (Deprecated) Qt5 CAN Trace View UI Service
 - CAN FD Trace View -> Qt5 CAN FD Trace View UI Service
 - DevTB2 Support Service -> ADTF 2.x Bus Data Deserializer Support Service
 - FlexRay Trace View -> Qt5 FlexRay Trace View UI Service
 - SOME IP Trace View -> Qt5 SOME IP Trace View UI Service
 - Display TB
 - 2D OpenGL Display -> Qt5 2D OpenGL Display
 - 3D OpenSceneGraph Display -> Qt5 3D OpenSceneGraph Display
 - Demo Signal Provider UI Service -> Demo Qt5 Signal Provider UI Service
 - Signal Scope View UI Service -> Qt5 Signal Scope View UI Service
 - Signal Table View UI Service -> Qt5 Signal Table View UI Service
 - Signal Tree View UI Service -> Qt5 Signal Tree View UI Service

#7 - 2022-05-18 15:36 - hidden

Hi Florian,

thanks for the detailed information! Now I think I get closer to understand the future concept.

For me it still looks a little bit hairy since names can be arbitrary changed in the graphs and the system. So how should the configuration editor or launcher know when opening a project with arbitrary named components, which Qt version is used? Hm, still we have the plugin descriptions and the path to the different plugins. So guess this can be handled somehow using that information.

Thank you! Ticket can be closed.

#8 - 2022-05-19 06:23 - hidden

- Project changed from 7 to Public Support
- Subject changed from On updating filters and service names to Prevent updating label names
- Description updated
- Status changed from Customer Feedback Required to To Be Closed
- Private changed from Yes to No
- Resolution set to Solved Issue
- Topic set to ADTF::Common

Hi Jochen,

For me it still looks a little bit hairy since names can be arbitrary changed in the graphs and the system. So how should the configuration editor or launcher know when opening a project with arbitrary named components, which Qt version is used? Hm, still we have the plugin descriptions and the path to the different plugins. So guess this can be handled somehow using that information.

The label name is only to differ in component view to select the correct one.

The CE creates the mapping label name <-> CID and knows the related plugin description, this is the ground truth and isn't change (and won't be).

So in the component view you have to entries

- Qt5 Media Description Display
 - pointing to demo_qt_media_description_display.ui_filter.adtf.cid
 - dependency qt_xsystem.osborn.services.adtf.iid (auto added)
- Qt6 Media Description Display
 - pointing to qt6_media_description_display.ui_filter.adtf.cid
 - dependency qt6_xsystem.osborn.services.adtf.iid (auto added)

The CE will check and decline using both XSystem Dependencies in one system while it will be added (manually as service or automatic as dependency).

In the Property Editor the user can figure out which component is used within the graph (no matter if renamed or not).

The Launcher uses Qt5 or Qt6 depending on the used XSystem within the System File.

And yes, having the same Session for both worlds means 2 graphs and two systems, which are not combinable.

Ticket can be closed.

OK

#9 - 2022-05-19 07:56 - hidden

Ah I see, new cids for the qt6 versions. Then it is clear.

BR,
Jochen

#10 - 2023-02-09 07:58 - hidden

- *Status changed from To Be Closed to Closed*