

Public Support - Support Request #11354

Migration help for 3D Mixin from ADTF 2.x to ADTF 3.x

2020-06-02 09:02 - hidden

Status:	Closed	Product Issue Numbers: Affected Products: ADTF 3.6.3, ADTF Display Toolbox 3.5.0 Platform: Windows 10 64bit Topic: DisplayTB::3DSceneDisplay FAQ Links:
Priority:	Normal	
Category:		
Customer:	AUDI	
Department:	AEV	
Requester's Priority:	Normal	
Support Level:	2nd Level	
Resolution:	No Customer Feedback	

Description

Supportanfrage

Is there any migration help/guidelines for the 3d mixins from adtf2 to adtf3?
I can see that there are main changes in the architecture of the mixins, beside the fact that OSG version has been also updated. Adding the changes in adtf3-framework itself to the equation, makes the migration of mixins a hard task (especially for mixins beginners like me!)

Changes which are difficult to map to the newer version:

- mixin callbacks have completely change
- Many types are not existent (e.g. cMixin::tPanelLayout, adtf_graphics::ISceneNodes ...)
- OSG migration help is difficult to find online (please mention a reference if you have?)
- ADTF3 IConfiguration behaves totally different, which was used in old mixins to set their properties

Lösung

First of all, just to be sure, please use ADTF 3.7.1 in combination with the Display Toolbox 3.5.0. It won't work with ADTF 3.6.3.

Yes, there have been many changes, between the major versions. The biggest architectural change is, that Mixins are no longer a special kind of plugin but are just regular ADTF Filters that expose their OSG parts via Interface Binding. The API (cMixin) hides this change from you, so I guess this is not to your concern.

Now addressing your points:

- mixin callbacks have completely change

We tried to keep them close to each other

- cMixin::OnPick is almost the same, only the node path argument is now an array instead of a (harder to use) IEnumObject (this also addresses your next point)
- cSensorManipulator has stayed the same

Or are you talking about the virtual functions of cMixin? Yes these have been adjusted to use the plain ADTF Filter Init() and Shutdown() functions which map to the old InitMixin() and ShutdownMixin() functions.

InitScene()/ClearScene() offer the same functionality as before.

- Many types are not existent (e.g. cMixin::tPanelLayout, adtf_graphics::ISceneNodes ...)

This is another architectural change. Since Mixins are just Filters they can create there own UI using the same mechanisms as all other Filters. Just use adtf::disptb::mixinlib::cQtMixin as you base class and then override the "QWidget* CreateView()" function and return your user interface there. The layout of the UI windows is then managed by the ADTF X System (the main UI window). So there no longer are any "panels", thus this type is no longer needed.

As mentioned above ISceneNodes is now replaced by a plain osg::Node array, that is much easier and straight forward to use.

- OSG migration help is difficult to find online (please mention a reference if you have?)

As far as I know we switched from OSG 3.2 to OSG 3.5 (so a minor version bump). This implies, that your "old" code should still be valid OSG code. I'm sure there are many subtleties, but a bump in the OSG version was highly requested by our customers. Unfortunately I don't know of any resources that address these issues.

- ADTF3 IConfiguration behaves totally different, which was used in old mixins to set their properties

If you're referring to the use of the property_variable<> template, then yes, but IConfiguration itself has actually not changed that much.

You can still set/get Properties manually as was the case in ADTF 2. But those functions are hard to discover, as these are free functions now:

- adtf::base::set_property<>() lets you set a property
- adtf::base::get_property<>() enables you to retrieve a property value.

Please have a look at our guide on properties (https://support.digitalwerk.net/adtf/v3/guides/sdk_properties_filter.html), that also shows the use of get/set_property.

I'm sorry, but there is no definitive guide for porting ADTF2 components to ADTF3, but we are very happy to help you with any questions on how to solve particular problem, so please do not hesitate to get back at us with further questions.

History

#1 - 2020-06-03 11:51 - hidden

- Topic set to DisplayTB::3DSceneDisplay

#2 - 2020-06-03 12:03 - hidden

- Status changed from New to In Progress

#4 - 2020-06-08 09:12 - hidden

Hi Samer Hanna,

sorry to hear that you're having difficulties porting your Mixins.

First of all, just to be sure, please use ADTF 3.7.1 in combination with the Display Toolbox 3.5.0. It won't work with ADTF 3.6.3.

Yes, there have been many changes, between the major versions. The biggest architectural change is, that Mixins are no longer a special kind of plugin but are just regular ADTF Filters that expose their OSG parts via Interface Binding. The API (cMixin) hides this change from you, so I guess this is not to your concern.

Now addressing your points:

- mixin callbacks have completely change

We tried to keep them close to each other

- cMixin::OnPick is almost the same, only the node path argument is now an array instead of a (harder to use) IEnumObject (this also addresses your next point)
- cSensorManipulator has stayed the same

Or are you talking about the virtual functions of cMixin? Yes these have been adjusted to use the plain ADTF Filter Init() and Shutdown() functions which map to the old InitMixin() and ShutdownMixin() functions.

InitScene()/ClearScene() offer the same functionality as before.

- Many types are not existent (e.g. cMixin::tPanelLayout, adtf_graphics::ISceneNodes ...)

This is another architectural change. Since Mixins are just Filters they can create their own UI using the same mechanisms as all other Filters. Just use adtf::disptb::mixinlib::cQtMixin as your base class and then override the "QWidget* CreateView()" function and return your user interface there. The layout of the UI windows is then managed by the ADTF X System (the main UI window). So there are no longer any "panels", thus this type is no longer needed.

As mentioned above ISceneNodes is now replaced by a plain osg::Node array, that is much easier and straight forward to use.

- OSG migration help is difficult to find online (please mention a reference if you have?)

As far as I know we switched from OSG 3.2 to OSG 3.5 (so a minor version bump). This implies, that your "old" code should still be valid OSG code. I'm sure there are many subtleties, but a bump in the OSG version was highly requested by our customers. Unfortunately I don't know of any resources that address these issues.

- ADTF3 IConfiguration behaves totally different, which was used in old mixins to set their properties

If you're referring to the use of the property_variable<> template, then yes, but IConfiguration itself has actually not changed that much.

You can still set/get Properties manually as was the case in ADTF 2. But those functions are hard to discover, as these are free functions now:

- adtf::base::set_property<>() lets you set a property
- adtf::base::get_property<>() enables you to retrieve a property value.

Please have a look at our guide on properties (https://support.digitalwerk.net/adtf/v3/guides/sdk_properties_filter.html), that also shows the use of get/set_property.

I'm sorry, but there is no definitive guide for porting ADTF2 components to ADTF3, but we are very happy to help you with any questions on how to solve particular problems, so please do not hesitate to get back at us with further questions.

Regards,

Martin

#5 - 2020-06-08 22:56 - hidden

- Status changed from In Progress to Customer Feedback Required

#7 - 2020-06-22 09:08 - hidden

- Project changed from 11 to Public Support
- Subject changed from Migration Help for 3D Mix from adtf2 to adtf3 to Migration help for 3D Mixin from ADTF 2.x to ADTF 3.x
- Description updated
- Status changed from Customer Feedback Required to To Be Closed
- Private changed from Yes to No
- Resolution set to No Customer Feedback

#11 - 2020-07-07 12:50 - hidden

- Status changed from To Be Closed to Closed