Public Support - Support Request #8232

ADTF plugin/filter versioning and migration to new version questions

2019-08-30 14:36 - hidden

Status:	Closed		
Priority:	Normal		
Category:			
Customer:	BOSCH	**Product Issue Numbers:**	
Department:		**Affected Products:**	ADTF 3.3.3, ADTF 3.6.1
Requester's Priority:	Normal	**Platform:**	Windows 10 64bit
Support Level:	2nd Level	**Topic:**	ADTF::Common
Resolution:	No Customer Feedback	**FAQ Links:**	

Description

Supportanfrage

I would like to ask for advice/help regarding the ADTF version 3.3.3.
We are building a system based on this version for quite a while and now emerged a need for the proper versioning of our filters. I checked the versioning macros in ADTF 3.3.3 but I could not even compile. We found the problem inside "adtf_version_customer" struct and in the ADTF_PLUGIN_VERSION and the ADTF_ADDITONAL_VERSION_TYPE macros.
What would you recommend to use instead of this (since this is not working)? Is there any alternative beside the dll VERSIONINFO or creating a dummy property? Currently a ADTF version change is not possible since the ADTF 3.3.3 is not compatible with 3.6.1 and multiple teams are working with this. Moving this many people could take months…

My other question would be that even if we would like to move to the new ADTF version

- Would it be a good idea?
- What are the responses from other teams?
- What are the plans for future releases and patches?
- Could we expect similar problems with it?
  I know it is hard to answer a question like this but when could we expect a stable release, which will worth the time to move?

Lösung

> I would like to ask for advice/help regarding the ADTF version 3.3.3.
> We are building a system based on this version for quite a while and now emerged a need for the proper versioning of our filters. I checked the versioning macros in ADTF 3.3.3 but I could not even compile. We found the problem inside "adtf_version_customer" struct and in the ADTF_PLUGIN_VERSION and the ADTF_ADDITONAL_VERSION_TYPE macros.
> What would you recommend to use instead of this (since this is not working)? Is there any alternative beside the dll VERSIONINFO or creating a dummy property?

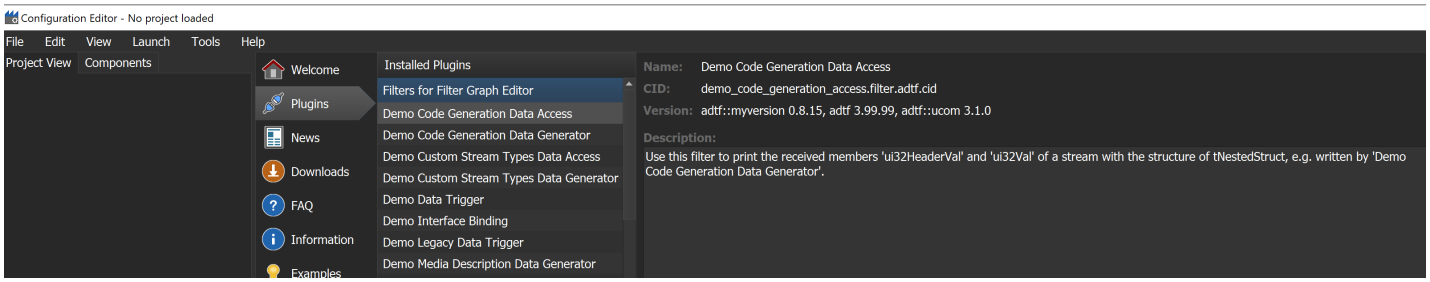
Keep in mind there has been a bug within ADTF 3.3.3, which is fixed in current delivery.
This works fine, just use ADTF_PLUGIN_VERSION instead of ADTF_PLUGIN.
If I extend one of our examples:

```
#include "demo_code_generation_access_filter.h"
#include "demo_code_generation_generator_filter.h"

/// this creates the plugin entry methods and class factories
ADTF_PLUGIN_VERSION("Demo Code Generation Plugin", myversion, 0, 8, 15,
            cCodeGenerationDataGenerator,
            cCodeGenerationDataAccess);
```

And regenerate the plugin description,

you will see in current Configuration Editor the exported and set customer version:



> Currently a ADTF version change is not possible since the ADTF 3.3.3 is not compatible with 3.6.1 and multiple teams are working with this. Moving this many people could take months…

The ADTF versions are definetly binary and code compatible along the whole ADTF 3 major version.
You can just use your binaries without recompiling within in the new ADTF version and it will work fine.
You can also recompile your code as well.

What do you mean by not compatible ?

Maybe your implementation is not explicit. Idea:
That means, you are using an cFilter (when you implemented was version namespace ::ant), with ADTF 3.5 there comes a new more helpful and convience Filter API (have a look at our guides and examples), but this is ::flash.
As long as you are using ant::cFilter explicit, your old implementation will work, but if you are using only cFilter it will be resolved to flash::cFilter and not work.
Also your trigger functions must still work, I guess only a namespace clash.
This could be an error, but only if you try to compile. If you are using binaries, everything will work (same as whole ADTF 2.x major version, ADTF is code and binary compatible through a major version).

But you have not to recompile, just yuse your old binaries and only imlement new ones with new Filter SDK, you will love.
If you are using trigger functions, they will be deprecated.
Also if you have problems with ADTF 3.3.3 we won't patch, there are so many good things, features and patches since ADTF 3.5

> My other question would be that even if we would like to move to the new ADTF version
> Would it be a good idea?

Definetly, yes, at least ADTF 3.5 (new and easier Filter SDK), we would recommend 3.6.x (tool redesign scripting, and stuff, just have a look at the release notes)

> What are the responses from other teams?

Great about new Filter SDK

> What are the plans for future releases and patches?

The main topics are publish subscribe for multiplexed data
SOME/IP
tooling automazation
more scripting filters
distributed system setups and comunication

for more details, please refer to your customer responsible (Roland Herrmann)

> Could we expect similar problems with it?

There are no problems for migration

> I know it is hard to answer a question like this but when could we expect a stable release, which will worth the time to move?

ADTF 3.5.0/3.6.x is much more stable than older releases

Related issues:

Related to Public Support - Support Request #8420: EBPRODUCTSUPPORT-4827 [EB ...	**Closed**

History

#1 - 2019-08-30 15:07 - hidden

- Project changed from Public Support to 5

- Topic set to ADTF::Common

#2 - 2019-08-30 15:44 - hidden

- File version.png added

- Status changed from New to Customer Feedback Required

Hi Gergely ,

> I would like to ask for advice/help regarding the ADTF version 3.3.3.
> We are building a system based on this version for quite a while and now emerged a need for the proper versioning of our filters. I checked the versioning macros in ADTF 3.3.3 but I could not even compile. We found the problem inside "adtf_version_customer" struct and in the ADTF_PLUGIN_VERSION and the ADTF_ADDITONAL_VERSION_TYPE macros.
> What would you recommend to use instead of this (since this is not working)? Is there any alternative beside the dll VERSIONINFO or creating a dummy property?

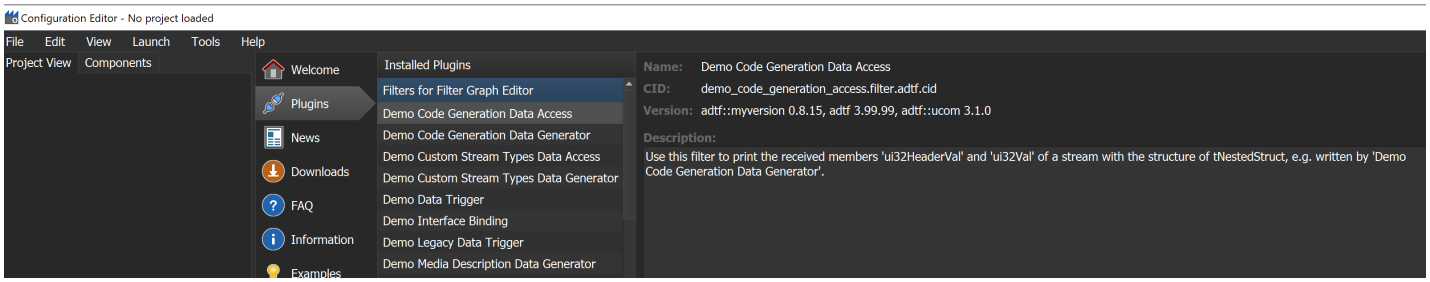
This works fine, just use ADTF_PLUGIN_VERSION instead of ADTF_PLUGIN.
If I extend one of our examples:

```
#include "demo_code_generation_access_filter.h"
#include "demo_code_generation_generator_filter.h"

/// this creates the plugin entry methods and class factories
ADTF_PLUGIN_VERSION("Demo Code Generation Plugin", myversion, 0, 8, 15,
            cCodeGenerationDataGenerator,
            cCodeGenerationDataAccess);
```

And regenerate the plugin description,

you will see in current Configuration Editor the exported and set customer version:



> Currently a ADTF version change is not possible since the ADTF 3.3.3 is not compatible with 3.6.1 and multiple teams are working with this. Moving this many people could take months…

The ADTF versions are definetly binary and code compatible along the whole ADTF 3 major version.
You can just use your binaries without recompiling within in the new ADTF version and it will work fine.
You can also recompile your code as well.

What do you mean by not compatible ?

Maybe your implementation is not explicit. Idea:
That means, you are using an cFilter (when you implemented was version namespace ::ant), with ADTF 3.5 there comes a new more helpful and convience Filter API (have a look at our guides and examples), but this is ::flash.
As long as you are using ant::cFilter explicit, your old implementation will work, but if you are using only cFilter it will be resolved to flash::cFilter and not work.
Also your trigger functions must still work, I guess only a namespace clash.
This could be an error, but only if you try to compile. If you are using binaries, everything will work (same as whole ADTF 2.x major version, ADTF is code and binary compatible through a major version).

But you have not to recompile, just yuse your old binaries and only imlement new ones with new Filter SDK, you will love.
If you are using trigger functions, they will be deprecated.
Also if you have problems with ADTF 3.3.3 we won't patch, there are so many good things, features and patches since ADTF 3.5

> My other question would be that even if we would like to move to the new ADTF version

> Would it be a good idea?

Definetly, yes, at least ADTF 3.5 (new and easier Filter SDK), we would recommend 3.6.x (tool redesign scripting, and stuff, just have a look at the release notes)

> What are the responses from other teams?

Great about new Filter SDK

> What are the plans for future releases and patches?

The main topics are publish subscribe for multiplexed data
SOME/IP
tooling automazation
more scripting filters
distributed system setups and comunication

for more details, please refer to your customer responsible (Roland Herrmann)

> Could we expect similar problems with it?

There are no problems for migration

> I know it is hard to answer a question like this but when could we expect a stable release, which will worth the time to move?

ADTF 3.5.0/3.6.x is much more stable than older releases

Again, as far as I imagine these are the only "problems" you can havebut easy to resolve.
If not, just let us know whats the problems.

If you require any additinal information on this topic, please let me know.

#3 - 2019-09-02 09:02 - hidden

- File 2019-09-02_08h20_41.png added

- File 2019-09-02_08h22_25.png added

- File 2019-09-02_08h27_55.png added

- File 2019-09-02_08h28_22.png added

- File 2019-09-02_08h29_44.png added

- File 2019-09-02_08h29_54.png added

Hello,

Have you tried the **ADTF_PLUGIN_VERSION** macro with ADTF 3.3.3?
The attached code segments are from the documentation, I will compare 3.3.3 and 3.6.1.
For the compatibility problem you can take this as an example too.
Yes I also had problems with a few namespace difference, but that isn't that crucial. I think a function being instance method in 3.3.3 then static method in 3.6.1 is a different level... Unfortunately I didn't tried to run already existing ADTF 3.3.3 filter with ADTF 3.6.1.

> But you have not to recompile, just yuse your old binaries and only imlement new ones with new Filter SDK, you will love.
> If you are using trigger functions, they will be deprecated."

Under compatibility problems I meant that we will have trouble recompiling our existing filters.
We are still developing our filters. "not recompiling" means that we could not develop further our already existing plugins. That would be unacceptable in our current status.

If the 3.6.1 proves itself being stable we will try to migrate to ADTF 3.6.1.
Is there any page of the that could help the migration except the release notes?

#4 - 2019-09-02 09:34 - hidden

Hi Bado,

> Have you tried the ADTF_PLUGIN_VERSION macro with ADTF 3.3.3?
> The attached code segments are from the documentation, I will compare 3.3.3 and 3.6.1.

Yes, you are right, there has been a bugfix, I forgot... sorry.

> Unfortunately I didn't tried to run already existing ADTF 3.3.3 filter with ADTF 3.6.1.

No problem, but this must work for sure.

> We are still developing our filters. "not recompiling" means that we could not develop further our already existing plugins. That would be unacceptable in our current status.

Sure, then you have to compile if you are still developing, this was just an option if its not your use case.

> Under compatibility problems I meant that we will have trouble recompiling our existing filters.

Then we must have a detailed look what is not working, each compilation error.
As long as you are using the old classes and namespaces (explicit), it must work, we did not remove deprecated methods or code.
If you are not working explicit, this could cause trouble.
So I can't give you a generic answer for what you are feeling like compatibility problems...

> Is there any page of the that could help the migration except the release notes?

The most important thing is getting confirm with the convinience API:

- Filter SDK

You will see, it feels more like doing the same for different components, getting confirm with different use cases und using same API for e.g. Filters and Streaming Services.

Then have a look at our basic examples and especially our guides

You will see how much easier and straight forward it will be to define components, pins, properties and stuff.
Using media Description (and header generation) and access the types (link).
And of course, more usability in tooling.

If you have any upcoming or detailed questions, just let us know.
We can also provide trainings or assist you using your components within the improved Filter SDK.

#5 - 2019-09-06 08:05 - hidden

- Project changed from 5 to Public Support

- Subject changed from ADTF plugin/filter versioning | migration to new version questions to ADTF plugin/filter versioning and migration to new version questions

- Description updated

- Private changed from Yes to No

- Resolution set to No Customer Feedback

#6 - 2019-09-06 08:05 - hidden

- Status changed from Customer Feedback Required to To Be Closed

#7 - 2019-09-23 14:32 - hidden

- Related to Support Request #8420: EBPRODUCTSUPPORT-4827 [EB Internal] ADTF 3.5 macro ADTF_PLUGIN_VERSION added

#9 - 2020-03-19 12:13 - hidden

- Related to Support Request #10807: Filter revision versioning in ADTF 3.x added

#10 - 2020-07-07 12:49 - hidden

- Status changed from To Be Closed to Closed

Files

version.png	77.9 KB	2019-08-30	hidden
2019-09-02_08h20_41.png	27.9 KB	2019-09-02	hidden
2019-09-02_08h22_25.png	28.4 KB	2019-09-02	hidden
2019-09-02_08h27_55.png	27.5 KB	2019-09-02	hidden

2019-09-02_08h28_22.png	29.4 KB	2019-09-02	hidden
2019-09-02_08h29_44.png	25.9 KB	2019-09-02	hidden
2019-09-02_08h29_54.png	25.2 KB	2019-09-02	hidden