From: ***<Team 1, Member 2>***
Sent: Tuesday, April 9, 2019 12:42 PM
To: ***Fayyazi, Morteza <…@mentor.com>***; ***<Team 1, Member 1>***
Cc: ***<Team 1, Member 2>***
Subject: Re: ***<Feature 1>*** integration

Sounds good. With regard to ***<Team 2 Feature 1 class API>*** map, I see two options

1. There is the keyword "mutable" that ***<Team 2, Level 1 Manager>***'s team can utilize on internal data structs that must change state when a const method is invoked.

2. Do as what they have already appeared to have done -- declare the method mutable (i.e. non const), but the returned output data const.

My opinion is that where ever there is output data structs returned, option 2 should be exploited. If there are simple query methods returning no output data, method 1 should be used as it avoids confusion to the user.

With regards,

<Team 1, Member 2>

From: ***Fayyazi, Morteza <…@mentor.com>***
Sent: Tuesday, April 9, 2019 12:20 PM
To: ***<Team 1, Member 2>***; ***<Team 1, Member 1>***
Subject: RE: ***<Feature 1>*** integration

Here ***Mr. Fayyazi, Morteza*** signaled that s/he asked <***Team 1, Member 1>*** to provide suggestion regarding the constness of the ***<Team 2, Feature 1 class API>***!!!!

After discussion with ***<Team 1, Member 1>***,
- He'll provide his suggestions for constness changes in ***<Team 2, Feature 1 class>***
- He'll remove ***<Team 1, Feature 1 API>*** from ***<Team 1, Feature 1 class>*** and provide it only in ***<Team 1, Feature 1 class>*** with two flavors:
1- Without any argument (return true ***<Mr. Fayyazi, Morteza provides detail about class internal>***)
2- With a string name for a ***<redacted text 6>***

Regards,
- ***Morteza***

From: ***<Team 1, Member 2>***
Sent: Tuesday, April 9, 2019 11:39 AM
To: ***<Team 1, Member 1>***; ***Fayyazi, Morteza <…@mentor.com>***
Cc: ***<Team 1, Member 2>***
Subject: Re: ***<Feature 1>*** integration

Here ***<Team 1, Member 2>*** recommends sending a patch that explains how to work around the constness issue highlighted earlier.

Hi ***<Team 1, Member 1>***,

Since there are multiple issues with the consistency and correctness of the ***<Team 2, Feature 1 class>*** map API, my suggestion is for you to put together a comprehensive diff that fixes all issues and send it for review and commit by ***<Team 2, member 1>***. The diff should be such that your code compiles clean without any casting hacks. I don't think ***<Team 2, Level 1 Manager>***'s team is going to ***<redacted text 7>***.

With regard to API ***<Team 1, Feature 1 API>***, simply provide me with the no input argument solution where this API is now in class ***<Team 1, Feature 1 class>*** instead of the ***<Team 1, Feature 1 class>***. This allows you to handle any multiple ***<Team 1, Feature 1 object name>*** case in your implementation should it ever become a requirement.

With regards,

<Team 1, Member 2>

From: ***<Team 1, Member 1>***
Sent: Tuesday, April 9, 2019 11:28 AM
To: ***<Team 1, Member 2>***; ***Fayyazi, Morteza <…@mentor.com>***
Subject: RE: ***<Feature 1>*** integration

I believe we are diverting the thread a bit, so attempt to combine them here.

For ***<Team 1, Feature 1 API>***, I have both APIs in place, so your pick.

For constness thingie, I spoke with ***<Team 2, Member 1>*** and s/he told me s/he would have a look if this is possible.

BTW, there was another API that caused me cast away constness and I wanted to change but had to hold back my breath:

```
// Get <Redacted test 1>
bool <Get Method Name>
(<Team 2 Feature 1 class>*,
 std::unordered_map<std::string, const <Team 2 Feature 1 class>*>&);

// Get <Redacted test 2>
bool <Get Method Name>
(<Team 2 Feature 1 class>*,
 std::unordered_map<std::string, const <Team 2 Feature 1 class>*>&);
```

The issue with the above APIs is that they are forcing me to cast constness away as on one hand the map returned contains constant pointer and on the other hand when I attempt to use those pointers with the same API I need to take constness away. However, I did not touch it and casted the constness away ☹

Here ***<Team 1, Member 1>*** voices concerns about the constness issue.

Cheers,
<Team 1, Member 1>

From: ***Fayyazi, Morteza <…@mentor.com>***
Sent: Tuesday, April 9, 2019 11:17 AM
To: ***<Team 1, Member 1>***; ***<Team 1, Member 2>***
Subject: RE: ***<Feature 1>*** integration

<Team 1, Member 1>,

Regarding 1, the significant of the change is not important.
I expect that you'll need a follow up commit after ***<Team 2, Member 1>***'s implementation anyway which you can remove const cast (in addition to enabling unit test).

Regarding 2, by relying on ***<Team 2, Feature 1 class>*** for ***<Team 1, Feature 1 API>***, you are already limiting your API to a single ***<Team 1, Feature 1 object name>***.
So it's the same as keeping this API in ***<Team 1, Feature 1 class>***, removing the argument, and internally using ***<Team 1, Feature 1 class method>***().
I believe that ***<Team 1, Feature 1 class>*** can be a good place for this API only if we process ***<Team 1, Level 1 Manager gives details about internal design>***.
With the current implementation (which decision is purely made based on optimizer) I don't see benefit of moving API to ***<Team 1, Feature 1 class>***.

Having said that, I don't have any strong preference on the place for this API. I leave it to you and ***<Team 1, Member 2>*** to agree which fits better for ***<Team 1, Feature 1 executable>***.

Regards,
- ***Morteza***

From: ***<Team 1, Member 2>***
Sent: Tuesday, April 9, 2019 11:17 AM
To: ***<Team 1, Member 1>***; ***Fayyazi, Morteza <…@mentor.com>***
Cc: ***<Team 1, Member 2>***
Subject: Re: ***<Feature 1>*** integration

My comments are as follows:

1. Since this change is so simple, I would think we could get easy approval from ***<Team 2, Level 1 manager>***'s team. In fact, this is a coding error on ***<Team 2, Member 1>***'s part for not declaring the method as being const. So my suggestion is to ask for approval, and commit change ourselves to avoid further red tape.

2. I don't mind ***<Team 1, Member 1>***'s suggestion here as it does not add any extra overhead or complications in ***<Team 1, Feature 1 executable>*** exe. I don't like having ***<Team 1, Feature 1 executable>*** to have to dig into the ***<Team 1, Feature 1 class>*** implementation to obtain the input arguments to method ***<Team 1, Feature 1 class method>***.

So, please let me know what is decided so I can make any necessary adjustments on my side.

Thanks, regards,

<Team 1, Member 2>

From: ***<Team 1, Member 1>***
Sent: Tuesday, April 9, 2019 11:16 AM
To: ***Fayyazi, Morteza <…@mentor.com>***; ***<Team 1, Member 2>***
Subject: Re: ***<Feature 1>*** integration

Hello,

Please find attached a patch that takes away constness from the <***Team 2, Feature 1 Get Attribute API***>. Also, added an <***Team 1, Feature 1 class method***> API that accepts an iterator.

--
Cheers,
<***Team 1, Member 1***>

From: ***<Team 1, Member 1>***
Sent: Tuesday, April 9, 2019 10:57 AM
To: ***Fayyazi, Morteza <…@mentor.com>***; ***<Team 1, Member 2>***
Subject: RE: ***<Feature 1>*** integration

Hi ***Morteza***,

Please find comments inline.

Cheers,
<Team 1, Member 1>

From: ***Fayyazi, Morteza <…@mentor.com>***
Sent: Tuesday, April 9, 2019 10:47 AM
To: ***<Team 1, Member 1>***; ***<Team 1, Member 2>***
Subject: RE: ***<Feature 1>*** integration

Hi ***<Team 1, Member 1>***,

My comments:

1- We should not change other teams code. I can request them if we believe some changes need to be done but they need to commit it.

Please plan your commit without any change in ***<Team 2, Feature 1 class>***.
[Team 1 Member 1] There will be unnecessary const cast in order to call non-constant method on a const pointer. I can do that, however I thought adding const should not be a big deal as it is not altering anything

2- The main use of <***Team 1, Feature 1 API***> is for <***Team 1, Feature 1 executable***> to decide if it should use <***Team 1, Feature 1 class A***> or not after <***Team 1, Feature 1 class B***>. Moving the API to <***Team 1, Feature 1 class A***> defeats the purpose.
I understand that your design assumes multiple <***Team 1, Feature class object name***> (although we don't have such a case today).
How about changing <***Team 1, Feature 1 API***> argument to get a pointer to the <***Team 1, Feature class object name***> rather than a string name?
i.e., <***Mr. Fayyazi, Morteza gives an API signature***>
[Team 1 Member 1] I can sure do so. However, I am not sure that having the API in <Team 1, Feature 1 class A> defeat any purpose. The <Team 1, Feature 1 class A> accepts two references to <Team 1, Feature 1 class C> and <Team 2, Feature 1 class> and it does not do any processing until Process is called. So, before calling process, the new API could then be called to let <Team 1, Feature 1 class A> identify if it would return so many <Team 1, Feature 1 objects> or not. It sounds to me like a natural place to be in. The caller can then proceed with Process call or avoid it. I can also provide the above API as an overloaded API.

Regards,
- ***<Team 1, Level 1 Manager>***

From: ***<Team 1, Member 1>***
Sent: Tuesday, April 9, 2019 8:40 AM
To: ***Fayyazi, Morteza <…@mentor.com>***; ***<Team 1, Member 2>***
Subject: Re: ***<Feature 1>*** integration

Thanks ***Morteza***,

I incorporate <***Team 2, Member 1***>'s change with <***Team 1, Feature 1 class***> modification. I just needed to make the API const. I am hoping I can get away with pushing const next to the API defined below. If not, I will need to perform const_cast which I am not favoring.

<***Team 1, Member 2***>,

As suggested, I added parameter-less <***Team 1, Feature 1 class method***>. However, I added it to the <***Team 1, Feature 1 class A***> class since it knows about both <***Team 1, Feature 1 class C***> and <***Team 2, Feature 1 class***> classes.

New patch attached. The new patch also includes the following:

1. Fix some misspelling or adding omitted words
2. Use of find_if instead of for_each when possible in the <***Team 1, Feature 1 class method***>() API
3. Had to disable the <***Team 1, Feature 1 mock class of Team 2, Feature 1 class***> int test as currently <***Team 1, Feature 1 mock class***> does return cUnknown for <***Team 1, Feature 1 object***>. Will enable once <***Team 2, member 1***> pushes changes that implement the new API

--
Cheers,
<***Team 1, Member 1***>

From: ***Fayyazi, Morteza <…@mentor.com>***
Sent: **Monday, April** 8, 2019 07:31 PM
To: ***<Team 1, Member 2>***; ***<Team 1, Member 1>***
Subject: RE: ***<Feature 1>*** integration

FYI, ***<Team 2, Member 2>*** has committed the change in ***<Team 2, Feature 1 class>***.h.

- ***Morteza***

From: ***Fayyazi, Morteza <…@mentor.com>***
Sent: Monday, April 8, 2019 4:34 PM
To: ***<Team 1, Member 2>***; ***<Team 1, Member 1>***
Subject: RE: ***<Feature 1>*** integration

Hi ***<Team 1, Member 1>***,

<Team 2, Member 1> is going to commit the requested API in ***<Team 2, Feature 1 class>*** for ***<Feature 1 attributes API>*** as described below. Please adjust your integration accordingly.

- ***bool <Team Feature 1 class>::<Get Attribute method>(<Attribute>&)***

  Returns **true** if the <***Team 2, feature 1 class has attribute***>, **false** otherwise. Retrieves the <***Team 2 Feature 1 class attributes***> if the <***Team 2 feature 1 class has attributes***> where <***Team 2 Feature 1 Attribute***> is an enumeration with the following definition:
  enum <***Attribute***> {<***Attribute 1***>, <***Attribute 2***>, <***Attribute 3***>, <***Attribute 4***>, unknown);
  An <***Attribute 1***> <***redacted text 1***>.
  An <***Attribute 2***> <***redacted text 2***>.
  An <***Attribute 3***> <***redacted text 3***>.
  An <***Attribute 4***> <***redacted text 4***>.
  If the <***Team 2, feature 1 class does not have attribute***>, then the <***Attribute***> returned is **unknown**. <***redacted text 5***>.

Here, ***Mr. Fayyazi, Morteza*** has channeled ***<Team 1, Member 1>*** API request and is echoing the response received from <***Team 2, Member 1***>!!!!

Thanks,
- ***Morteza***

From: ***<Team 1, Member 2>***
Sent: Monday, April 8, 2019 4:02 PM
To: ***<Team, Member 1>***
Cc: ***Fayyazi, Morteza <…@mentor.com>***; ***<Team 1, Member 2>***
Subject: Re: ***<Feature 1>*** integration

Thanks ***<Team, Member 1>***.

--***<Team 1, Member 2>***

From: ***<Team 1, Member 1>***
Sent: Monday, April 8, 2019 3:59 PM
To: ***<Team 1, Member 2>***
Cc: ***Fayyazi, Morteza <…@mentor.com>***
Subject: RE: ***<Feature 1>*** integration

Hey ***<Team 1, Member 2>***,

Will do! Expect a new patch soon.

Cheers,
<Team 1, Member 1>

From: ***<Team, Member 2>***
Sent: Monday, April 8, 2019 3:51 PM
To: ***<Team 1, Member 2>***
Cc: ***Fayyazi, Morteza <…@mentor.com>***
Subject: Re: ***<Feature 1>*** integration

Hi ***<Team, Member 1>***,

I believe method ***<Team 1, Feature 1 class method>*** on class ***<Team 1, Feature class>*** should be defined to have no input arguments. This method should internally ***<Team 1, Member 2 describes class internals>***. This avoids ***<Team 1, Member 2 gives details on the benefits>***.

Thanks, regards,

<Team 1, Member 2>

From: ***<Team 1, Member 1>***
Sent: Monday, April 8, 2019 9:51 AM
To: ***<Team 1, Member 2>***
Cc: ***Fayyazi, Morteza <…@mentor.com>***
Subject: RE: ***<Feature 1>*** integration

Hi ***<Team 1, Member 2>***,

Yes, ***<Team 1, Feature 1 mock class>*** is an old class and will go with this patch. The patch would reference only ***<Team 1, Feature 1 class same as Team 2 Feature 1 class>*** class.

Cheers,
<Team 1, Member 1>

From: ***<Team 1, Member 2>***
Sent: Friday, April 5, 2019 5:40 PM
To: ***<Team 1, Member 1>***
Cc: ***Fayyazi, Morteza <…@mentor.com>***; ***<Team 1, Member 2>***
Subject: Re: ***<Feature 1>*** integration

Hi ***<Team, Member 1>***,

One point: Why does class ***<Team 1, Member 2 asks question about an old mock class>***.

Thanks, regards,

<Team 1, Member 2>

From: ***Fayyazi, Morteza <…@mentor.com>***
Sent: Friday, April 5, 2019 4:44 PM
To: ***<Team 1, Member 1>***; ***<Team 1, Member 2>***
Subject: RE: ***<Feature 1>*** integration

Hi ***<Team 1, Member 1>***,

The changes look very good.
Please wait until I confirm if the push can be done.

Regards,
- ***Morteza***

Here ***Mr. Fayyazi, Morteza*** signal his OK for the change to go in! Also, s/he points out that ***<Team 1, Member 1>*** API requested as been channeled to ***<Team 2, Level 1 Manager>***'s team.

From: ***<Team 1, Member 2>***
Sent: Friday, April 5, 2019 01:25 PM
To: ***<Team 1, Member 1>***
Cc: ***Fayyazi, Morteza <…@mentor.com>***; ***<Team 1, Member 2>***
Subject: Re: ***<Feature 1>*** integration

How long is it going to take ***<Team 2, Level 1 Manager>*** to respond?

From: ***<Team 1, Member 1>***
Sent: Friday, April 5, 2019 1:23 PM
To: ***<Team 1, Member 2>***
Cc: ***Fayyazi, Morteza <…@mentor.com>***
Subject: RE: ***<Feature 1>*** integration

Morteza told me to wait on ***<Team 2, Level 1 Manager>***'s team approval as I am pushing the following into their files:

```
class <Some Feature 1 Class>
{
public:
  enum <Attribute 1, 2, and 3>;

public:
  .
  .
  .

  <Get Attribute API>;
};
```

Note: This is an implicit requested for an API that ***Mr. Fayyazi, Morteza*** channeled to ***<Team 2, Level 1 Manager>*** in another email thread that ***<Team 1, Member 1>*** is not part of!!!!

Along two other spots where ***<The above API is defined in subclasses>***.

Cheers,
<Team 1, Member 1>

From: ***<Team 1, Member 2>***
Sent: Friday, April 5, 2019 1:18 PM
To: ***<Team 1, Member 1>***
Cc: ***Fayyazi, Morteza <…@mentor.com>***; ***<Team 1, Member 1>***
Subject: Re: ***<Feature 1>*** integration

typo -- ***<Team 1, Member 1 corrects some typos>***.

Thanks, regards,

<Team 1, Member 2>

From: ***<Team 1, Member 2>***
Sent: Friday, April 5, 2019 1:16 PM
To: ***<Team 1, Member 1>***
Cc: ***Fayyazi, Morteza <…@mentor.com>***; ***<Team 1, Member 2>***
Subject: Re: ***<Feature 1>*** integration

Go ahead and commit your latest patch. I don't see anything fundamentally wrong, just some cleanups that can be addressed as a follow-on. The goal is for me to be able to link ***<Redacted text>***.

Thanks, regards,

<Team 1, Member 2>

From: ***<Team 1, Member 1>***
Sent: Friday, April 5, 2019 1:12 PM
To: ***<Team 1, Member 2>***
Cc: ***Fayyazi, Morteza <…@mentor.com>***
Subject: RE: ***<Feature 1>*** integration

In the meantime, you can apply my patch to your branch if you may.

Cheers,
<Team 1, Member 1>

From: ***<Team 1, Member 1>***
Sent: Wednesday, April 3, 2019 11:21 AM
To: ***<Team 1, Member 2>***
Cc: ***Fayyazi, Morteza <…@mentor.com>***
Subject: RE: ***<Feature 1>*** integration

Hey ***<Team 1, Member 2>***,

This method could be used by client to report any ***<Redacted text 1>***. It could be used to alert users that those ***<Redacted text 2>***.

Cheers,
<Team 1, Member 1>

From: ***<Team 1, Member 2>***
Sent: Wednesday, April 3, 2019 11:17 AM
To: ***<Team 1, Member 1>***
Cc: ***Fayyazi, Morteza <…@mentor.com>***; ***<Team 1, Member 2>***
Subject: Re: ***<Feature 1>*** integration

Thanks ***<Team 1, Member 1>***.

Regarding your ***<Feature 1 class>*** class, what is the purpose of method ***<Feature 1 class method>***?

--***<Team 1, Member 2>***

From: ***<Team 1, Member 1>***
Sent: Wednesday, April 3, 2019 10:40 AM
To: ***<Team 1, Member 2>***; ***Fayyazi, Morteza <…@mentor.com>***
Subject: Re: ***<Feature 1>*** integration

Hi ***<Team 1, Member 2>***,

Thanks for your comments.

Yes the patches integrates with ***<Team 2, feature 1>*** APIs as stated earlier in the brief description below. The mock classes are required for unit testing which I intend to preserve.

Regarding the ***<Feature 1 class>*** name, I prefer simple class names myself. Namespaces could be used to convey the meaning that it is ***<Feature 1 class>*** specific. Since this is not part of this commit, I will defer to a separate commit once we agree on a resolution.

Attaching new patch that addresses the following:

1. Adding new test cases that uses ***<Team 2 feature 1 APIs>***. ***<Generated Data from existing test case>*** is checked in along the test case
2. Address a bug (or better say missing logic) that handles square brackets as regex set. They needed to be escaped which I missed earlier. Test for this bug is part of the above test case.

--
Cheers,
<Team 1, Member 1>

On Tue, 2019-04-02 at 18:20 -0400, ***<Team 1, Member 2>*** wrote:
Hi ***<Team 1, Member 1>***,

I see that you are linking with ***<Team 2, Level 1 manager>***'s ***<Feature 1>*** library. This is good news! Disregard my first comment. Can we now get rid of the mock source code in your ***<Feature x>*** directory?

Thanks, regards,

<Team 1, Member 2>

From: ***<Team 1, Member 2>***
Sent: Tuesday, April 2, 2019 4:28 PM
To: ***<Team 1, Member 1>***
Cc: ***Fayyazi, Morteza <…@mentor.com>***; ***<Team 1, Member 2>***
Subject: Fw: ***<Feature 1>*** integration

You should also rename your class ***<Class name 1>*** to ***<Class name 2>*** to be explicit that it extracts ***<Redacted text 3>*** from the expressions provided in a ***<Redacted text 4>***.

Thanks, regards,

<Team 1, Member 2>

From: ***<Team 1, Member 2>***
Sent: Tuesday, April 2, 2019 3:59 PM
To: ***<Team 1, Member 1>***
Cc: ***Fayyazi, Morteza <…@mentor.com>***; ***<Team 1, Member 2>***
Subject: Re: ***<Feature 1>*** integration

Hi ***<Team 1, Member 1>***,

Other than the missing ***<Get Attribute>*** function, what is holding you back from using ***<Team 2, Level 1 Manager>***'s ***<Feature 1>*** API implementation in your ***<Feature 1>*** extract code right now?

If we can't use the real API that ***<Team 2, Level 1 Manager>*** has provided, then I suggest you move the mock class implementations that your code currently depends on from the ***<Feature x>*** directory to your ***<Feature 1>*** directory under ***<Redacted text>***. This code should then be archived in the ***<Internal module>*** library archive so that it is linkable with the ***<Feature 1>*** exe that I am delivering in my next commit. Your unit tests should then be updated to get the respective .h files from the ***<internal module>*** source tree and should also link with ***<internal library>***.

Once you fully integrate ***<Team 2, Level 1 Manager>***'s ***<Get Attribute>*** function, you can then remove the mock source files.

Thanks, regards,

<Team 1, Member 2>

From: ***<Team 1, Member 1>***
Sent: Tuesday, April 2, 2019 9:13 AM
To: ***Fayyazi, Morteza <…@mentor.com>***
Cc: ***<Team 1, Member 2>***
Subject: RE: ***<Feature 1>*** integration

Hi ***Morteza***,

The change in the patch sent earlier is related to:

"Integrating ***<Feature 1 related attribute>*** extract algorithm with ***<Team 2, Feature 1>*** APIs (commit: `a5b5df6fba33c9a76181942bb18153c56385673d`.) Also, changes the existing test case mock class to incorporate ***<Team 2>*** naming conventions for classes and APIs. Mock class implements ***<Team 2>*** APIs to obtain ***<Feature 1 related attributes>***. There is no logic change, just adopting ***<Team 2>*** API style."

Cheers,
<Team 1, Member 1>

From: ***Fayyazi, Morteza <…@mentor.com>***
Sent: Tuesday, April 2, 2019 9:07 AM
To: ***<Team 1, Member 1>***
Cc: ***<Team 1, Member 2>***
Subject: FW: ***<Feature 1>*** integration

Hi ***<Team 1, Member 1>***.

Please send a description of the changes.
I've added ***<Team 1, Member 2>*** also to review.

Thanks,
- ***Morteza***

From: ***<Team, Member 1>***
Sent: Monday, April 1, 2019 1:55 PM
To: ***Fayyazi, Morteza <…@mentor.com>***
Subject: ***<Feature 1>*** integration

Hi ***Morteza***,

Please find attached patch for integrating <***Team 2 Feature 1 API***> into <***Team 1, Feature 1***> class.

My sandbox is: <***internal dir***>
branch: <***git branch***>
--
Cheers,
<***Team 1, Member 1***>