From: ***<Team 1, Member 1>***
Sent: Monday, April 15, 2019 09:16 AM
To: ***Fayyazi, Morteza <…@mentor.com>***; ***<Team 2, Member 2>***
Cc: ***<Team 1, Team 2>***; ***<Team 1, Level 1 Manager>***
Subject: Re: ***<Feature 1 Get Attribute>*** API

Mr. Fayyazi, Morteza is thanking ***<Team 2, Member 1>*** for work for which ***Mr. Fayyazi, Morteza*** decided to fire ***<Team 1, Member 1>*** For!!! Hilarious, right!!!

Thanks a lot ***<Team 2, Member 1>***!

Will pull the change in.

--
Cheers,
<Team 1, Member 1>

On Sun, 2019-04-14 at 22:27 -0400, ***Fayyazi, Morteza <…@mentor.com>*** wrote:
Thanks ***<Team 2, Member 1>***.

It is ironic here that ***<Team 2, member 1>*** pushed code into ***<Team 1>***'s code tree without receiving same backlash ***<Team 1, Member 1>*** received for asking the question to push into ***<Team 2>***'s code tree!!!

On Apr 14, 2019, at 8:46 PM, ***<Team 2, Member 1>*** wrote:

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

I pushed the ***<Feature 1 Get Attribute>*** implementation to ***<git branch>***.
This push included a change to `bool` ***`<New feature 1 API>`***`()`
to align with a request from the ***<Team 4>*** team.

In doing so I had to change some const qualifications in the ***<Team 1, Feature 1 Class>*** code,
which I think were what you had specified in your email.
Please let me know if there are any issues with this.

Regards,
<Team 2, Member 1>

On 04/10/2019 03:11 PM, ***<Team 1, Member 1>*** wrote:
The previous patch did not have the mutable change I spoke of below. Now it is attached to this email.

--
Cheers,
<Team 1, Member 1>

On Wed, 2019-04-10 at 11:44 -0700, ***<Team 1, Member 1>*** wrote:
Hi ***<Team 2, Member 1>***,

Bringing this to the top of mailboxes.

Also. I had another observation regarding get ***<Team 2 Feature 1>*** APIs:

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

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

The above APIs forced me to cast constness away as ***<Second parameter>*** map has const pointers and those pointers are later on used to get their ***<first parameter>*** and so on. It is a no no for me when using API and cast away constness. I understand that the API is doing lazy loading of ***<Redacted text 3>***.

I thought about bubbling down the const cast which I did in the attached batch, however commented out. It is one liner change. However, I met resistance from my team and was recommended to use mutable modifier which I also did in the attached batch and is the active code. In either case, the final API now looks like:

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

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

The attached batch also modifies ***<internal module>*** code and remove the const_cast. The code is simple and I have no problem if you so choose to push that code along other that introduce const to API argument. However, I'd leave that for management to decide.

--
Cheers,
<Team 1, Member 1>

From: ***<Team 1, Member 1>***
Sent: Tuesday, April 9, 2019 09:01 AM
To: ***<Team 1, Member 2>***
Cc: ***Fayyazi, Morteza <…@mentor.com>***; ***<Team 1, Member 2>***; ***<Team 2, Level 1 Manger>***
Subject: : ***<Feature 1 Get Attribute>*** API

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

Thanks for make the change to incorporate ***<Feature 1 Get Attribute>*** API. One minor change I would like to push along my integration patch is to make the API const, so I would like to change it to be:

```
virtual bool <Get Attribute API>(…) const { Attribute = cUnknown; return true; }
```

If that is OK with you, that would be great and would proceed with push change once all green lights are turned on.

--
Cheers,
<Team 1, Member 1>