Subject: Re: Callback (THISBACK) Improve
Posted by tojocky on Mon, 20 Dec 2010 16:06:13 GMT
View Forum Message <> Reply to Message

kohait00 wrote on Fri, 17 December 2010 12:01i found out that the last change didnt really work and started to scratch Callback layer and found out many inconsistancies in name and usage.. so the past day i was restructuring the code blocks and renaming what was needed.

also, the changes recently made did not support the in-callback type conversion, i.e specifying a callback for float but parametrizing it with int.. this is corrected.

further more, i have extended Callback and Gate and unified their definitions.

since the code grew quite a bit with all that, i have grouped the stuff by count of arguments passing to THISBACK. so you will find a Callback0.h for THISBACK stuff, Callback1.h for THISBACK1, etc..
Callback.h simply includes them all..

just see the code. i think it's pretty clean now and easy to extend if needed, since all Callbacks now have same respective template type names accordingly. i havent changed the code in terms of function, but some naming issues to be more clean, i.e.

```
struct CallbackMethodActionArg1_2 : public Callback2Action<P1, P2> {}

//to

struct Callback2MethodActionArg1 : public Callback2Action<P1, P2> {}
```

because they made name clashes when extending..

for testing, i have compiled a major application, and also TheIDE, which both still work and also created a test app where all the combinations are listed to check right compilation. there, you will find the new added combinations and those still not supported.

i also added a STDBACK, STDBACK1, etc. helper, when dealing with non THISBACK functions (because i always have to remeber the syntax for those..)

EDIT:

there are now versions to reduce a callback's attributes by one, specifying a parameter from right, yiedling a callback with one parameter less..

i also found some templates for non thiscall's that use class R template type for return value, whereas callbacks are supposed to be void returning. is this a remanent of old times or is it 'by design'?

Nice work!