Subject: Re: using Ctrl::Add; required for templates / overloaded virtual functions
Posted by mrjt on Thu, 24 Jun 2010 08:51:24 GMT
View Forum Message <> Reply to Message

I see your problem. I think you may have run up against the limit's of the C++ compiler there. Congratulations

Of course the real problem is the manner used for overloading the Add(Ctrl &) method. If a method is going to be overloaded it should be done properly. For instance, this code works exactly as you would like:

```
struct Base {
 int result;
 virtual void Add(Ctrl &) { result = 1; }
};

struct Mid : public Base{
 virtual void Add(Ctrl &) { result = 2; }
 void Add(int)   { result = 4; }
};

struct Top : public Mid{
 void Nothing()  { result = 0; }
};


GUI_APP_MAIN
{
 Label l;
 Top md;

 //option 1: << desired >>
 void (Top::* mfp)(Ctrl &) = &Top::Add;
 (md.*mfp)(l);
}
```

The function pointer calls Mid::Add(Ctrl &).

So either the Ctrls that overload Add(Ctrl &) should be changed (surely they can use ChildAdded?) or Add(Ctrl &) should be made virtual.