Subject: Re: iTools ambiguity
Posted by Michael Galloy on Wed, 14 Jan 2009 15:46:50 GMT
View Forum Message <> Reply to Message

On Jan 13, 10:48 pm, Robbie <ret...@iinet.net.au> wrote:
> The documentation for IDLitDataContainer says that its superclasses
> are IDLitContainer and IDLitData however I attempt the following
>
> obj = obj_new('IDLitDataContainer')
> obj -> IDLitContainer::Add, obj_new('IDLitComponent')
>
> and I get an error. Instead I must do
>
> obj -> _IDLitContainer::Add, obj_new('IDLitComponent')
>
> Evidently the IDLitContainer behaviour has been implemented using
> _IDLitContainer (an abstract class).
>
> This is a somewhat loaded question, but which superclass is correct?
> Is it _IDLitContainer or is it IDLitContainer?

Looking at the source code it appears that _IDLitContainer is the real superclass that implements the "add" method. IDLitContainer is just a thin wrapper around _IDLitContainer (and also subclassing IDLitComponent).

The reason for all those _ classes is multiple inheritance: classes can't inherit from IDLitComponent twice (the dreaded diamond of death in the inheritance hierarchy). In this case, IDLitDataContainer inherits from IDLitData (a subclass of IDLitComponent) and _IDLitDataContainer. If it inherited from IDLitDataContainer directly then it would be an error, so an _IDLitContainer class is created which is all the functionality needed in IDLitContainer, but that doesn't inherit from IDLitComponent.

So the documentation is telling a "white lie" here, IDLitDataContainer doesn't inherit from IDLitContainer, but it has all the functionality as if it did. Of course, depending on what you are doing, this might really throw you off since it is not true.

There are many of the _ classes, I think generally they implement all the functionality and then are inherited by a class without the _ that also inherits from IDLitComponent.

This is what you get when you use multiple inheritance.

Mike
--

www.michaelgalloy.com
Tech-X Corporation
Associate Research Scientist