
Subject: Re: IDL objected oriented question

Posted by [Pavel Romashkin](#) on Wed, 09 Apr 2003 16:33:37 GMT

[View Forum Message](#) <> [Reply to Message](#)

There is no question that IDL object implementation is crippled. Just as there is no question that even as it is, it is very useful.

I have been following the threads addressing heap cleanup and this one, and - what can we do? - yes, we have to take care of many things that other languages do on their own.

With regard to Catalyst, I suppose, you just right click on [dfanning.com/catalyst.zip](#) and choose "download to disk", and enjoy :-)

Jokes aside, I thought of doing a similar thing as Catalyst, but for a different reason - using small widget systems that don't even have to be visible to help objects do their business. So to say, fill the gaps in object implementation with widget events, just like in, for instance, VBA, where objects can simply listen to each other (in addition to true encapsulation and automatic cleanup :-)

Then of course, it all comes back to a global event sink, object self-awareness and transmogrification. Which ends up to be a Common variable or elusive orphaned pointer :-)

The reason I haven't done it is, I never really needed it. Shouldn't be very difficult and a fun, challenging project. Any takers?

Cheers,

Pavel

- > Dave Burrige and I have solved this problem with our Catalyst
 - > Object Library by wrapping all widgets up as objects. Then widget
 - > events automatically get sent to event handler *methods* rather
 - > than event handler procedures. This makes it possible to write
 - > widget programs in the normal way, but you get to take advantage
 - > of the many lovely properties of objects, too. It is the best
 - > of both worlds, really.
-