## Subject: Re: Class Warfare
Posted by Michael Wallace on Fri, 06 Aug 2004 07:07:29 GMT

View Forum Message <> Reply to Message

```
>  Here's an example... First create a Java string.
>
>  IDL> session = Obj_New('IDLjavaObject$IDLJAVABRIDGESESSION')
>  IDL> jstring = Obj_New('IDLjavaObject$java_lang_String', $
>                         'java.lang.String', 'My String')
>
>  OBJ_ISA will tell you that it is a string, but doesn't tell you that it
>  is also an Object even though java.lang.String directly inherits from
>  java.lang.Object.
>
>  IDL> print, Obj_Isa(jstring, 'IDLjavaObject$java_lang_String')
>     1
>  IDL> print, Obj_Isa(jstring, 'IDLjavaObject$java_lang_Object')
>     0
```

I don't know if anyone else is interested in this stuff, but I've learned another interesting IDL-ism.  Just thought I'd pass it on just in case there are other Java programmers besides me out here.

It looks like I was expecting way too much out of Obj_Isa().  Obj_Isa() doesn't really do much of anything with Java objects.  When a Java object gets created, the part of the name following 'IDLjavaObject$' appears to be purely arbitrary.  I can enter any string I want to in that area, and it's the string that Obj_Isa() appears to use for the so-called instanceof operator.

```
; Create a Java string, but give a random string for the object name
IDL> jstring = Obj_New('IDLjavaObject$java_lang_Foo', $
                       'java.lang.String', 'My String')

; Obj_Isa() is matching the string provided...
IDL> print, Obj_Isa(jstring, 'IDLjavaObject$java_lang_Foo')
   1

; And Obj_Isa() doesn't match the actual type of the class!
IDL> print, Obj_Isa(jstring, 'IDLjavaObject$java_lang_String')
   0
```

So, the answer to my original question is to give an appropriate name to my objects when I create them.  If I will only be used an interface of a specific object, I can create the name to match the interface name and viola! Obj_Isa() will return what I expect.  The only limitation is that for each object created, you can only choose one class or interface. I still think a true instanceof operator needs to exist. I'd rather handle class typing problems by checking instanceof beforehand rather than relying on the error handler to tell me I have a type problem.

-Mike