

---

**Subject: Class Warfare**

Posted by [Michael Wallace](#) on Wed, 04 Aug 2004 21:31:00 GMT

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

---

I'm using several Java classes in my IDL code and I'm attempting to make the code a little more robust by doing some simple type checking of the Java objects I'm sending into various procedures and methods. What I'm after is something which does the same thing as the Java instanceof operator. The OBJ\_ISA function gets part of the way, but doesn't do an entire instanceof check like Java does.

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
```

In some situations there is a workaround. I can create a new Java Object and get this object's Class and use that to test the Java string.

```
IDL> jobject = Obj_New('IDLjavaObject$java_lang_Object', $
    'java.lang.Object ')
IDL> jobjcls = jobject -> GetClass()
IDL> print, jobjcls -> IsInstance(jstring)
1
```

While this works in a pinch, there are situations where this can't be done to easily. There are classes which take time and/or resources to instantiate and it doesn't make sense to instantiate a class only to check if another class is an instance of it and not do anything else with the class.

For the time being, I'm using OBJ\_ISA with the exact class that I expect, but it'd be much nicer to have some way to have a true Java instanceof operator.

-Mike

---