Subject: Asynchronous IDL_IDLBridge causing memory leak
Posted by Seth Johnson on Fri, 27 Aug 2010 18:09:27 GMT
View Forum Message <> Reply to Message

Hello,

I have recently be toying with the IDL_IDLBridge object to try to perform multiple simultaneous operations. In my testing, however, I have noticed that running a command asynchronously, through use of the NOWAIT keyword in the IDL_IDLBridge::Execute method, will result in a memory leak over a large number of iterations. A simple example of this can be re-created using:

```
a=bindgen(1E4,1E3)

oBridge=Obj_New('IDL_IDLBridge')
oBridge->SetVar,'a',a
FOR i=0,999 DO BEGIN
   tmp=memory()
   oBridge->Execute,'a=a+a',/NOWAIT
   print,memory(/high)
   WHILE oBridge->Status() NE 0 DO wait,0.0001
ENDFOR
```

Removing the NOWAIT keyword shows no additional memory usage on successive iterations. I realize that including the WHILE...DO... statement effectively makes the asynchronous process a moot point; however, my intention is to have multiple similar processes running at once and have the main IDL routine wait until all have finished, not have it wait on a one-by-one basis; e.g.:

```
oBridge=OBJARR(5)
FOR i=0,999 DO BEGIN
  FOR chain=0,4 do BEGIN
     oBridge[chain]=Obj_New('IDL_IDLBridge')
     oBridge[chain]->SetVar,'a',a
     oBridge[chain]->Execute,'a=a+a',/NOWAIT
  ENDFOR

  FOR chain=0,4 DO WHILE oBridge[chain]->Status() NE 0 DO wait,0.0001
ENDFOR
```

I was wondering if there was perhaps a fix that could correct the leaking memory or if there might be an alternative to initializing and running multiple IDL child processes. This bug is particularly annoying as the memory is not freed when the IDL_IDLBridge objects are destroyed, it just sits in the parent IDL session with the only available option for retrieving the lost memory is to exit and restart

the IDL session.

I appreciate any advice and help regarding this issue.