Subject: Using Callable IDL/ActiveX component to access variables
Posted by Mark McGillion on Fri, 15 Jan 1999 08:00:00 GMT
View Forum Message <> Reply to Message

Hi,

I am using callable IDL from a C++Builder application on a WindowsNT (x86) platform. I have embedded the IDLDrawWidget ActiveX component within a form and I am successfully starting up IDL and plotting to the IDLDrawWidget component. To do this, I use the IDLDrawWidget.ExecuteStr() function and a .PRO file containing additional functions and procedures.

My problem is that I cannot access the variables contained within my IDL .PRO file. I have several COMMON global variables that I need to access. I am trying to use the IDLDrawWidget.GetNamedData(char*) function which should return a VARIANT data type that references the IDL variable.

However, the return value is UNDEFINED. When I examine the help files, I cannot find a reference to this function. Instead, a reference is made to another function IDL_FindNamedVariable(char*, int) that returns an IDL_VPTR that references the variable.

Question:
1. Am I using the correct function to access the data variable?
2. Why does the help file contain info on a completely different function?
3. I am currently not linking the IDL32.DLL to my application. However, when I examine the DLL, I see that it contains the IDL_FindNamedVariable(char*, int) function, but not the GetNamedData(char*) function. It is possible that I am getting confused between the activeX component functions and the DLL functions?
4. Should I be using the DLL functons as well/instead of the activeX component functions?

Here is my C++ code:

```
TIDLDrawWidget *IDLDrawWidget1;

void __fastcall TMainForm::ButtonAnalysisClick(TObject *Sender)
{
   float *Data;
   char *var;
   Variant ptr;

   var = new char[100];   //space to hold the variable name
```

```
    var = "C_buffer";         //IDL COMMON variable

    Data = new float [1000];

     ptr = IDLDrawWidget1->GetNamedData(var);
//    or should I use IDL_VPTR ptr = IDL_FindNamedVariable(var,
FALSE);

       Data = (float*) ptr;
       AnalysisForm->AnalyseData(Data);
}
```

Please help as I cannot find a way out of this seemingly simple problem.

Sincerely yours

Mark
