
Subject: Dynamically creating C++ objects in DLM
Posted by [Rick Towler](#) on Sat, 21 Dec 2002 00:21:45 GMT
[View Forum Message](#) <> [Reply to Message](#)

It has been a slow day here so maybe this can stir up some passion :)

I have a few .dlm's where I declare a global C++ object and use that object throughout the dlm. The dlm functions and procedures allow me to initialize the object, do stuff with it, and destroy it. Easy enough.

The one limitation is that I only have one instance of the object available. If I run a second instance of a program that uses that .dlm the two IDL programs will overwrite the C++ object causing general mayhem (like using a common block in IDL).

A solution to this problem would be dynamically creating the C++ object upon initialization and returning a pointer to the object back to IDL. When calling the dlm routines I could pass the pointer back to my dlm to gain access to my object of interest.

My questions are:

How do I dynamically create a C++ object (I think I use "new" but am a little unclear on the correct use)?

```
myObject *test = new myObject; //?????
```

How do I pass pointers back and forth between my DLM and IDL? Is it as simple as passing a long?

```
return IDL_GetTmpLong(test); //Is it this easy?
```

If I do pass IDL a long representing my pointer, how do I use it when I pass it back to my DLM? The compiler sees it as a long, not a pointer to an object.

```
pointer = IDL_LongScalar(argv[0]); //receive the pointer address from  
IDL  
*pointer.Test(); //not the way to do it
```

Thanks for any thoughts.

-Rick
