

Re: beginner Q's about COM/.NET (context = exposing a pre-existing app to scripting)

## Re: beginner Q's about COM/.NET (context = exposing a pre-existing app to scripting)

---

*Source:* <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.atl/2006-01/msg00270.html>

---

- *From:* "Brian Muth" <[bmuth@xxxxxxxx](mailto:bmuth@xxxxxxxx)>
  - *Date:* Tue, 24 Jan 2006 08:49:51 -0800
- 

I'll give a stab, but I'm sure others will chime in.

"Jason S" <[jmsachs@xxxxxxxx](mailto:jmsachs@xxxxxxxx)> wrote in message  
[news:1138113287.763127.220220@xx](mailto:news:1138113287.763127.220220@xx)

- > Hello, I am not sure if this group is the best place to post but I have
- > found it very unclear where the dividing lines are between
- > ATL/COM/.NET/Automation/ActiveX etc.
- >
- > I have an existing app which has a few classes (which I need to
- > re-architect more sanely, but that's besides the point). These are all
- > non-GUI, relatively simple; the only OS-related thing about them is
- > some of them use the serial port, operate in more than one thread &
- > therefore contain mutexes to access the serial port.
- >
- > They are and will continue to be implemented in C++ for various
- > reasons. At present I am starting out using ATL in MSVC 6.0.
- >
- > I am trying to figure out how to wrap COM around (or incorporate it in,
- > or whatever you call it) these classes to expose them to scripting
- > languages & am having trouble finding a decent tutorial; all the
- > tutorials out there on the 'net seem to be too simple to explain some
- > of the common & important subtleties, or they're about GUI objects which
- > are different in a lot of ways. (e.g. I have no interest in object
- > linking/embedding)
- >
- > I understand most of the basic aspects of COM but am struggling with a
- > few of the subtleties.
- >
- > Are there good resources to learn the following:
- >
- > a) Should I abandon COM & go for .NET?

Scripting languages, at the moment, only can deal with COM objects. You can create a COM wrapper (called an Interop) for a .NET object, so this doesn't mean you can't program in .NET. Overall, you will probably find .NET easier once you get the hang of it. The downside is that there will be a noticeable delay when the .NET object is first loaded, since compilation of the .NET assembly is done on the fly.

Re: beginner Q's about COM/.NET (context = exposing a pre-existing app to scripting)

Re: beginner Q's about COM/.NET (context = exposing a pre-existing app to scripting)

- > I can't even find a good site
- > explaining what .NET is and why it's different. I don't think we have
- > an updated compiler here that supports .NET & am reluctant to upgrade.
- > A software engineer friend at first told me it would be easier to work
- > with .NET, not COM, but then he said if I'm just doing basic things &
- > not worrying about inter-computer security or object
- > marshalling/storage then COM is probably fine.
- >
- > b) If I have a pre-existing object hierarchy where objects are created
- > within the hierarchy itself, how do I handle reference counting? I'm
- > fairly sure I won't be asking for new objects through CoCreateInstance
- > except for one initial call which returns a singleton object A; all the
- > other objects are derived through A's methods. e.g. B =
- > A.createB('B\_name') or C = A.lookupC('preexisting\_C'). B and C will
- > have cyclical links back to their parents & will not be destroyed
- > except explicitly through external methods. So it doesn't seem to make
- > sense to me how reference counting fits into the picture. (especially
- > how to handle the cyclical link case)

On the client side, reference counting is looked after automatically by the scripting language. As long as you have a reference to a COM object, you know the object has been AddRef'ed. When the reference goes out of scope or is set to NOTHING, then a call to Release() is implied.

By the way, in the COM world, you should not implement singleton objects. (This differs from the .NET world where singletons can work quite well). However, it is pretty easy to mimic singleton behaviour by using global variable or static class member variables to store the common information (for example, serial port information).

- >
- > c) what is the proper behavior for the COM server to take if a script
- > stores a reference to an object B, which no longer exists, and then the
- > script later calls one of B's methods? (which Murphy's law says WILL
- > happen)

Actually, it won't happen. As long as the scripting language is holding a reference, you know the COM object's reference count won't drop to zero and the object will continue to exist.

- >
- > d) is there a way to disallow CoCreateInstance on a particular class B?
- > It needs to be public because my scripts will be receiving instances of
- > class B from class A's methods, calling their methods & passing them as
- > parameters to class A's methods — but in no case do I want someone to
- > create an orphan class B outside the context of class A.

Yes. Objects can be marked as non-creatable.

>

Re: beginner Q's about COM/.NET (context = exposing a pre-existing app to scripting)

Re: beginner Q's about COM/.NET (context = exposing a pre-existing app to scripting)

- > e) best practices for passing exceptions back to the script clients
- > calling my COM methods

Look up ISupportErrorInfo, and AtlReportError

- >
- > f) recommended ways to create & return strings that contain formatted numbers -- right now I'm using a fixed char[] buffer, calling \_snprintf() on it, then copying the buffer to a \_bstr\_t, and storing it in the return value's variant BSTR field.

Sounds ok with me.

- >
- > g) is there a suggested COM architecture to provide mutex functionality? As I experiment w/ COM I am running a .EXE COM with the STA model... If client process X calls A.lockserialport(), then does stuff, then A.unlockserialport() and client process Y also makes the same sequence of calls on the same object, I'm not clear whether my COM server code would be running these in the same thread or different threads.

This gets into the notion of apartments. If your COM object is marked as "apartment-threaded", it lives in a single-threaded apartment (STA) and any calls to methods are serialized for you automatically. In this situation, you don't need a mutex at all. On the other hand, if your COM object is marked as "free-threaded", then it lives in a multi-threaded apartment (MTA) and multiple methods may be active at the same time. In this situation, access to common resources such as the serial port will need to be synchronized, and a mutex is one way of doing it.

- > If the server runs in the same thread via a message loop, a standard mutex would not block. If the server runs them in different threads, how do I guarantee this, and can I use WaitForSingleObject() or do I have to use MsgWaitForSingleObject()?

You will almost never to write a message loop in a simple COM object. See my previous comment.

HTH

Brian

.

---

• *Follow-Ups:*

◆ *Re: beginner Q's about COM/.NET (context = exposing a pre-existing app to scripting)*

◇ *From: Jason S*

Re: beginner Q's about COM/.NET (context = exposing a pre-existing app to scripting)

- **References:**

- ◆ [beginner Q's about COM/.NET \(context = exposing a pre-existing app to scripting\)](#)

- ◆ From: Jason S

- Prev by Date: [Re: ATL COM with IIS6 \(auto ExitInstance\)](#)

- Next by Date: [Re: Modbus opc server](#)

- Previous by thread: [beginner Q's about COM/.NET \(context = exposing a pre-existing app to scripting\)](#)

- Next by thread: [Re: beginner Q's about COM/.NET \(context = exposing a pre-existing app to scripting\)](#)

- Index(es):

- ◆ [Date](#)

- ◆ [Thread](#)