

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/msg00277.html>

- *From:* "Jason S" <jmsachs@xxxxxxxxxx>
 - *Date:* 24 Jan 2006 15:08:38 -0800
-

yes that helps. (d), (e), (f) I'm all set on, I think. I've got some books arriving tomorrow that should help more, I hope.

Brian Muth wrote:

> > 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.

hmm... (interested but skeptical) Would I be correct in assuming .NET is "easier" (less manual labor) to implement objects but has more conceptual baggage than COM? If so, I'll probably stick w/ COM. I'm an EE and only a portion of my job is software-related... the less complexity in what I do in this area, the better. I find that the MS tools to make things "easier" seem to work well when you do something simple, but as soon as you try to do something real which doesn't quite fit the mold you have to spend much more work understanding the tools and overriding their default options. The ATL stuff seems to fit this pattern to some degree.

> By the way, in the COM world, you should not implement singleton objects.

?

I found a webpage that cited putting

```
DECLARE_CLASSFACTORY_SINGLETON(Cmyclass)
```

in the declaration for Cmyclass. It seems to work as expected. Am I missing a subtlety here? (there's no warning in MSDN's docs for this macro)

> > 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

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)

- > reference, you know the COM object's reference count won't drop to zero and
- > the object will continue to exist.

does this mean if a script stuffs a reference to one of my server's objects into a script variable, and lets it sit on the shelf, my server is required to honor that request? I understand the philosophy... in my case I have an object referring to information that exists in hardware which may be unplugged, at which point (from a practical sense) the object ceases to be useful. I guess I could always free up the object's resources & mark it as invalid, even if the object itself continues to exist forever.

- >> 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.

I was assuming existence of a message loop handled "under the hood" by the ATL stuff.

I think I have to use MTA; while program A is using the serial port, I want program B to be able to check if the serial port is locked or unlocked, and that method call will block if I use STA on the same object. (right?) I'm just not clear yet on the caveats of MTA... there must be some shared resources I don't know about that I have to synchronize manually...

--Jason

.

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

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

- **References:**
 - ◆ **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)**
 - ◇ From: Brian Muth

- Prev by Date: **Re: Smart Type**
- Next by Date: **Re: beginner Q's about COM/.NET (context = exposing a pre-existing app to scripting)**
- Previous by thread: **Re: 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**