

microsoft.public.dotnet.framework.remoting: Re: 1st hosting of objects, new(), etc.

## Re: 1st hosting of objects, new(), etc.

**Source:**

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.remoting/2004-08/0251.html>

---

**From:** Sam Santiago (*ssantiago\_at\_n0spam-SoftiTechure.com*)

**Date:** 08/17/04

Date: Tue, 17 Aug 2004 14:20:43 -0700

In the server application any object you create with New will be a local object. If you wanted to create an reference to a remotely available object you would use the Activator.GetObject method. Also, be sure to override the InitializeLifetimeService method on your remote object that should have inherited from MarshalByRefObject to return nothing so that you have a true singleton.

You might want to check out some sample code I posted here:

<http://www.softitechure.com/discussions> regarding factory objects. Also, there was a discussion regarding factories in this newsgroup a couple of weeks ago that might be useful:

<http://groups.google.com/groups?hl=en&lr=&ie=UTF-8&threadm=%23ms6%24x%23dEHA.720%40TK2MSFTNGP>

<http://groups.google.com/groups?hl=en&lr=&ie=UTF-8&threadm=eACbPVXeEHA.332%40TK2MSFTNGP09.phx.g>

Thanks,

Sam

--

---

Sam Santiago  
ssantiago@n0spam-SoftiTechure.com  
<http://www.SoftiTechure.com>

---

"Mike" <vimakefile@yahoo.com> wrote in message  
news:%23FXIDsJhEHA.2764@TK2MSFTNGP11.phx.gbl...  
> I'm less interested in either app hosting the object, as I am in having  
the  
> IIS application object holding a reference to a speedy/local/non-proxy  
> reference (it is the host, but also the main client), and having any other  
> apps have proxies into the IIS-app-hosted object.  
>  
> So it sounds like I can do this by having a singleton factory of my own,  
> call it to get the a local reference, and then call registerServerType -  
no?  
> Or does it not make a difference when I call rWKST because on the server  
all  
> copies will be local -- e.g., there won't be a loopback socket that

Re: 1st hosting of objects, new(), etc.

microsoft.public.dotnet.framework.remoting: Re: 1st hosting of objects, new(), etc.

```
> eventually calls back into the same process.
>
> mike
>
> "Sam Santiago" <ssantiago@n0spam-SoftiTechure.com> wrote in message
> news:eLuZDbJhEHA.2764@TK2MSFTNGP11.phx.gbl...
> > The New statement will create a local object on the server
application
> > even if you have published a wellknown object of that type. On the
client
> > application once you have registered a wellknown type in that app domain
> > it
> > makes sense that New would attempt to create a remote object.
> > Your issue is that either one of your applications can act as a
> server.
> > You might want to reconsider that. I think it is a bit risky to have an
> > ASP.Net application dependent on a remote object in a Windows Forms
> > application that can be shut down at will. With that said you might
want
> > to
> > try something like the pseudo code below in each application, I am not
> > sure
> > if it would work:
> >
> > obj1 = New Myobject - Create a local instance of the object
> > bUseLocalObj = true
> > Register MyObject as a wellknown type
> > try
> >     obj2 = attempt to create another instance of your object (obj2 -
> should
> > be remote instance)
> >     bUseLocalObj = false
> > catch
> >     if error, remote object is not available, probably not been
published
> >     Marshal obj1 to a URI
> > end try
> >
> > if not bUseLocalObj then obj1 = obj2      (obj1 and obj2 are both
Myobject
> > types, but obj2 is a proxy, so not sure what would happen)
> >
> > Use obj1 throughout your application      (may or may not be a proxy based
> > on
> > what happened above)
> >
> >
> > Good luck.
> >
> > Thanks,
> >
> > Sam
> >
> >
> > --
> > _____
> > Sam Santiago
> > ssantiago@n0spam-SoftiTechure.com
> > http://www.SoftiTechure.com
> >
> > "Mike" <vimakefile@yahoo.com> wrote in message
> > news:eRS1Nz$gEHA.3272@TK2MSFTNGP11.phx.gbl...
```

## microsoft.public.dotnet.framework.remoting: Re: 1st hosting of objects, new(), etc.

```
> > > I'd like to have the first-running application that references my
object
> > > host a singleton server for the app's lifetime, and I'd like the
hosting
> > > application to use a local (non-remoted, same process) version of the
> > > object.
> > > For instance, if my winforms app is the first app to need the object,
it
> > > will do a registerWellKnownServerType on a singleton that is also used
> > > locally, and if an IIS application on the same machine runs first,
I'll
> > > create a the local object and cache it in the ASP.Net Application
object
> > and
> > > then register it. Otherwise, if one or the other is running, it will
use
> > a
> > > remoted version. Is it possible to have this
> > > local-and-remoted-access-to-the-same-singleton scenario work? (With
> > > appropriate lock()'s in case the local and remote context threads muck
> > with
> > > state at the same time.)
> > >
> > > The impetus is that I'd like a management console to be able to run
w/o
> > IIS
> > > active, but most of the time I want the IIS application to host the
> > object
> > > and have fast, local access to it. In the case that I run one or more
> > > managment consoles from local or remote machines with IIS already
> > hosting
> > > the object, they will connect to the already running object.
> > >
> > > One problem is .Net's choice of implicit remote creating via new()
which
> > I
> > > don't really like. It seems like before a registerWellKnownClientType
> > "new
> > > foo(...)" means one thing, and after the registration it means
something
> > > else - an explicit Remote.Create( typeof(foo), ...) seems cleaner.
Once
> > > rWKCT is called, how do I create local, non-remote, non-singleton
> > versions
> > > of
> > > the object if I wanted to?
> > >
> > > I guess in the case where the hosting app exits, it would cause an
> > > exception
> > > on the client, or I could have the process stick around while there
are
> > > still non-timed-out clients (unless IIS gets aggressive with an
> > > ostensibly
> > > zombified process.)
> > >
> > >
> > > thanks,
> > > mike
> > >
> > >
> > >
> > >
> > >
> > >
```

microsoft.public.dotnet.framework.remoting: Re: 1st hosting of objects, new(), etc.

>