

# Re: Using client side proxies to instantiate remote objects

**Source:**

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.remoting/2004-07/0004.html>

---

**From:** Richard Bell ([richardb\\_at\\_rogge.co.uk](mailto:richardb_at_rogge.co.uk))

**Date:** 06/30/04

Date: Wed, 30 Jun 2004 12:26:20 +0100

Thanks Sunny. I think (hope) we then agree. There are bugs which need workarounds. This particular bug, however, still does not require you to use the interface method.

The binaries are larger for all configurations of the interface method simply because you need more classes (actual and definition) and the client proxy assembly is meta data only. In fact, I believe, that the size of the distributed binaries can in some cases be up to 4 times larger (one touch SmartClients??). However, I do not see binary size as a major issue. I still see the more important issues as architectural. If you want your code to be truly reusable, location transparency should be (is) a design goal

However, the most critical problem in my mind is still versioning. Using the interface method, the client simply does not know the version of the server to which it would like to bind. It only knows the version of the 'shared' assembly. Unless I am mistaken, the client will bind to whichever version of the server is exposed at the end point. You have less versioning control than you did in DCOM where you could specify the version of the CoClass to which you wished to bind.

In my mind, client side meta data proxies are analogous to CoClasses. Using client side proxies, the client can be built to reference a specific version of the server assembly. However, which version of the server that actually binds can be configured at runtime by a network administrator using the system policy. All of this out of the box!

I do feel that as SmartClients become the norm, this issue has the ability bite back. I'll keep our coding policy of using client side proxies and perhaps we should compare notes in a year's time! Anyway, thanks again for replying. Rgds.

"> <http://www.genuinechannels.com/Content.aspx?id=88&type=1>

>

> *My tests so far prove that Dmitry is right, and this problem exists. So*  
> *one way or another, as you say, if there is a bug, we work around it,*

microsoft.public.dotnet.framework.remoting: Re: Using client side proxies to instantiate remote objects

- > *and I have to avoid config files and stick with programmatic*
- > *configuration of remoting. And the overcoding for this is exactly one*
- > *and the same for using the "new" or interface approach. Which kind of*
- > *simplifies the use of interface/base class approach, as it does not*
- > *require external tool.*
- >
- > *So for now I'll stick with my favorite and maybe in next release of the*
- > *framework, if still remoting is as good option as it is now, and with*
- > *better IDE and soapsuds I may reconsider my position.*
- >
- > *Sunny*
- >