

## Re: interprocess comms in local network

**Source:**

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2004-12/2641.html>

---

**From:** guy (*wildfiction\_at\_hotmail.com*)

**Date:** 12/10/04

Date: Fri, 10 Dec 2004 15:30:00 -0500

Hi Ian,

Many thanks for your replies and help a few weeks ago. I got side tracked on another project but now I'm back on this one.

Has anyone written an article comparing the different methods of interprocess communication across the internet.

My objective is to write a client/server application.

They won't be very complicated but will do this:

**Server**

Connects to a live data feed (via Excel) and retrieves data.

Does some calculations on the data.

Distributes data to clients on remote machines on internet.

Server needs to be an application with buttons and menus as it will need to be manipulated while it's running.

**Client**

Displays the data it receives.

Also an application with menus etc.

I also need to set up permissioning such that only clients that I've authorized can connect to the server.

Is there a template of this sort of application that I can use as a starting point.

Many thanks

Guy

"Ian Griffiths [C# MVP]" <ian-interact-sw@nospam.nospam> wrote in message news:OV4XJGK0EHA.2192@TK2MSFTNGP14.phx.gbl...

> "guy" <wildfiction@hotmail.com> wrote:

>> *Since writing I've been thinking that at some point I may want this*

>> *application to break out of the local area network and be available on*

>> *remote machines across the internet.*  
>>  
>> *I'm thinking that ASP.NET might give me this. Is this right?*  
>  
> *Yes, although in principal Remoting could too. You can configure a*  
> *remoting channel to listen on any port you like. And there's an HTTP*  
> *channel available too if that's the only thing that will tunnel through*  
> *your firewall.*  
>  
> *But the web services route would probably be a better solution here. The*  
> *reason I say that is that if you are deploying applications out on the*  
> *internet, you probably want to use a technology that is as robust as*  
> *possible in the face of changing requirements – you're probably not going*  
> *to have the luxury of being able to update all the clients simultaneously*  
> *when you update the server.*  
>  
> *Not that you get this kind of versioning–robustness for free using web*  
> *services, it's just easier to achieve.*  
>  
>  
>> *On a small local area network do I need to set something special up to*  
>> *use ASP.NET? You mentioned Web Services. Is this a service that is*  
>> *available on XP that just needs to be started?*  
>  
> *ASP.NET provides support for implementing a web service. (Specifically,*  
> *the server half of it.) The ASP.NET framework is installed as standard*  
> *with the .NET Framework, so if you've got the .NET Framework, you've got*  
> *ASP.NET.*  
>  
> *However, you need a host process in which to run ASP.NET in order to use*  
> *it. The usual way of doing this is to use the standard IIS–based hosting.*  
> *If IIS was installed when you installed the .NET Framework, it will have*  
> *set up the standard IIS–based hosting of ASP.NET. If it wasn't install*  
> *IIS, and then run this command:*  
>  
> *aspnet\_regiis -i*  
>  
> *You'll find that command in C:\windows\microsoft.net\framework\v1.1.4322.*  
> *This configures IIS to allow ASP.NET applications to be hosted. (This*  
> *adds an ISAPI extension that directs requests with certain extensions*  
> *through to an ASP.NET worker process. This process is called*  
> *aspnet\_wp.exe and is started up on demand when you first hit an ASP.NET*  
> *resource via IIS. Unless you're on Windows Server 2003, in which case*  
> *it'll call the process wpw3.exe.)*  
>  
> *Once you've made sure ASP.NET is registered with IIS, you can then create*  
> *a web service. For example, put a file in \inetpub\wwwroot and give it a*  
> *".asmx" extension, e.g. "MyService.asmx". Make it something like this:*  
>  
> *<%@ WebService language="c#" class="MyService" %>*  
>

```
> using System.Web.Services;
>
> public class MyService
> {
> [WebMethod]
> public int Add(int x, int y)
> {
> return x + y;
> }
> }
>
>
> You have now exposed MyService.Add as a web service. Try looking at the
> http://localhost/MyService.asmx file (or whatever you chose to call it) in
> your web browser, and it will generate some documentation telling you how
> to use the web service.
>
> You don't have to host via IIS though. The ASP.NET framework is happy to
> be hosted in any process. For example, this:
>
> http://www.iunknown.com/000099.html
>
> shows how to host ASP.NET from inside of a Windows Forms application.
>
>
> If you're using VS.NET, you can get it to generate you a wrapper class to
> consume a web service from the client side.
>
>
> --
> Ian Griffiths - http://www.interact-sw.co.uk/iangblog/
> DevelopMentor - http://www.develop.com/
>
```