

Remoting server auto-discovery. Broadcast or not?

Asked 16 years, 4 months ago Modified 15 years, 9 months ago

Viewed 4k times



4



I have a client/server application that communicates with .Net remoting. I need my clients to be able to find the server(s) on the network without requiring client-side configuration.

As far as I know, there is no support for discovery in remoting. I am prepared to implement a UDP solution where the clients will be able to find the servers via broadcast messages.

I want to hear the collective SO wisdom before I go ahead. Is this the best way? Any other suggestions?

.net

network-programming

remoting

udp

broadcast

Share

edited Aug 22, 2008 at 13:34

Improve this question

Follow

asked Aug 22, 2008 at 13:22



Ishmaeel

14.4k ● 9 ● 73 ● 84

5 Answers

Sorted by:

Highest score (default)



4



I've looked at both SSDP and UPnP for this type of functionality, but I'd recommend going with a custom UDP [multicast](#) solution. Basically, multicast is very similar to a broadcast, but only machines that have joined the multicast group (i.e. requested the broadcast) are contacted.

IMHO, SSDP and UPnP are bloated and overly complicated for resource discovery... but hey, it's a standard. ;)

Share Improve this answer

answered Oct 2, 2008 at 1:01

Follow



[oz10](#)

158k ● 27 ● 98 ● 129



1



Seems like what you need is the [Simple Service Discovery Protocol](#) or SSDP. This is implemented in Windows as part of Microsoft's support for [Universal Plug and Play](#). Since this is an industry standard protocol, it seems like a good bet. For instance, if you want to deal with firewalls or other issues, this will have been figured out by others instead of you having to roll your own solution.

Since you are talking .NET I'll assume you are on Windows. There's a somewhat old document (2001) describing a C-style API and a COM API for Windows

entitled [Universal Plug and Play \(UPnP\) Client Support](#).

The COM APIs are exposed by UPNP.DLL and the C-style APIs for SSDP are exposed by SSDPAPI.DLL.

The [COM-style APIs for UPNP](#) are probably your best bet. Since C# can wrap up COM objects for you and handle the interop. I could not find any place where this API has been ported to C# or the .NET Framework natively.

Share Improve this answer

answered Oct 1, 2008 at 22:19

Follow



[Tim Farley](#)

11.9k ● 4 ● 31 ● 30



1

You might also consider Apple's [Bonjour](#), which is their [Zeroconf](#) implementation. It's available for Mac, PCs, and Linux/BSD.



Share Improve this answer

answered Oct 2, 2008 at 1:13

Follow



[Will Hartung](#)

118k ● 20 ● 133 ● 207



0

The best solution I have found in my remoting work was to keep the server list in a config file on the client systems and make it updateable. Not the easiest to maintain but was fast and no broadcasting.



Share Improve this answer

answered Aug 22, 2008 at 13:36





Follow



Rob Bazinet

1,020 ● 1 ● 8 ● 12

I already have a client-side config solution in place. This no-touch requirement has newly dropped in my lap. Basically, we need to deploy to a massive amount of dummies end users with minimum administrative intervention. Thx anyway.

– [Ishmaeel](#) Sep 7, 2008 at 19:49



My multicast UDP solution seems to be unreliable due to recent MS update.

0

Share Improve this answer

answered Mar 25, 2009 at 6:43



Follow



Rohit

3,638 ● 7 ● 46 ● 76

