

Generic deserialization

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.remoting/2004-06/0560.html>

From: Lee Gillie (*ANTISPAMIFICATION_lee_at_odp.com*)

Date: 06/29/04

Date: Tue, 29 Jun 2004 16:07:29 -0700

Trying to create a "generic" notification facility. Basically I have a number of services which currently use NT MAILSLOTS to broadcast their in-progress status to desktops on the LAN. In the mailslot there is usually complex delimited data as a string.

My .NET approach is to instead use a "message server service". Also attempting to provide a component that will make either SOURCE or CONSUMER objects. The SOURCE object has one simple "Send" method, and accepts a parameter of type "Object". Am making a "Consumer" object which posts an event, and again, a single parameter of type OBJECT. I am attempting to hide all the infrastructure and remoting as much as possible, from the caller of the component. The component is also exposing a COM interface. The component makes either a source or consumer conversation with the service via one single remoted component, which can "broadcast" events to clients. All of this works great when the notify parameter is a string. But falls flat when trying to do this with an item of type OBJECT.

Here is my extreme lack of knowledge about serialization...

Via the remoting method, I was able to send from a SOURCE to the server, and it makes it back to the CONSUMER in a separate process. But the consumer is having trouble looking at the "object". I get this:

Run-time exception thrown :

System.Runtime.Remoting.RemotingException - This remoting proxy has no channel sink which means either the server has no registered server channels that are listening, or this application has no suitable client channel to talk to the server.

The "object" being sent through this system is a "class" using only fundamental data types, such as string, integer, and so on. The class is marked serializable, and I found a need to inherit from MarshalByRef to get the send side to work.

My thought is that once I get this class instance serialized, it is completely self contained. Yet the message seems to indicate the CONSUMER is somehow trying to get back to some remoting host, or such. Probably due to the MBR.

Any hints on preparing my generic "object" parameter to be fully self-sufficient would be greatly appreciated. In my component I don't have knowledge of my caller's classes. If I can, I would like to not burden my caller to perform serialization and deserialization, if possible.

Can what I am trying to do even be done?