Exercise 4

3.2. From the server side in project eLibraryServer: Defining class DAL for remote access.

Step 1

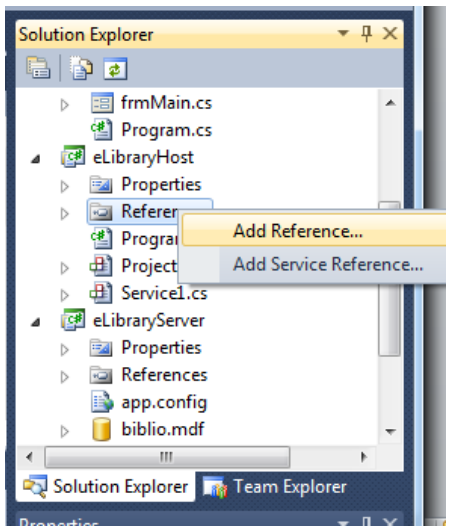
Add inheritance in class DAL - *marshal-by-reference*

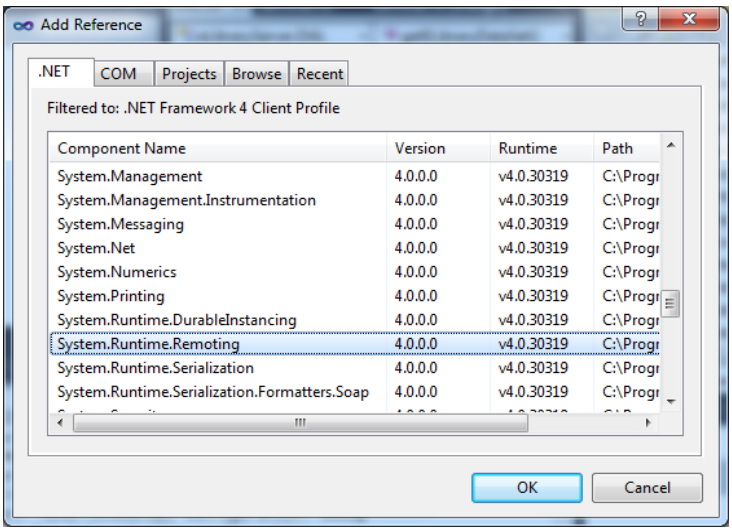
```
public class DAL : MarshalByRefObject
{
  …
public DataSet getELibraryDataSet()
{
  …
}
}
```

3.3. Project eLibraryHost: Configure remote access.

Step 1

- Add the following reference **System.Runtime.Remoting** (tab **".NET").**





- Add reference to the project eLibraryServer from tab Projects.
- Add reference to the project eLibraryClient from tab Projects.

Step 2

Edit file Service1.cs :

- Add the following libraries:

```
using System.Runtime.Remoting;
using System.Runtime.Remoting.Channels;
using System.Runtime.Remoting.Channels.Tcp;
```

- Create a function **doRemotingConfiguration**() with commands for configuring remote access.
- Call *doRemotingConfiguration* in *OnStart() function.*

```
using System.Runtime.Remoting;
using System.Runtime.Remoting.Channels;
using System.Runtime.Remoting.Channels.Tcp;

public partial class Service1 : ServiceBase
{
  protected override void OnStart(string[] args)
  {...
    doRemotingConfiguration();
  }
  private void doRemotingConfiguration()
  {
      TcpServerChannel channel;
      //-- create and register a TCP server channel on port 8089
      channel = new TcpServerChannel(8089);
      ChannelServices.RegisterChannel(channel, false);
      //-- register the service for remote access in SingleCall mode
      RemotingConfiguration.RegisterWellKnownServiceType(typeof(eLibraryServer.DAL), "eLibraryServer.rem", WellKnownObjectMode.SingleCall);
  }
```

3. 4. From the client side in the project: Connecting to a remote object

Step 1

- Add the following reference **System.Runtime.Remoting** (tab **".NET")**

Step 2

In the form "frmMain":

- Add the following libraries:

```
using System.Runtime.Remoting;
using System.Runtime.Remoting.Channels;
using System.Runtime.Remoting.Channels.Tcp;
```

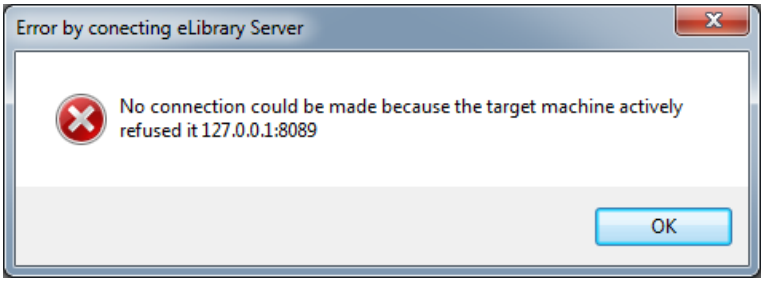
- Create a function **doClientRemotingConfiguration**() with commands for configuring remote access.
- Call *doClientRemotingConfiguration* in ***frmMain_Load***()

```
using System.Runtime.Remoting;
using System.Runtime.Remoting.Channels;
using System.Runtime.Remoting.Channels.Tcp;

public partial class frmMain : Form
{    ...
private void frmMain_Load(object sender, EventArgs e)
 {    ...
      doClientRemotingConfiguration();
 }
```

```
private void doClientRemotingConfiguration()
{
 TcpClientChannel channel= new TcpClientChannel();
 //-- Register a TCP client channel

 ChannelServices.RegisterChannel(channel, false);
 //-- Register the remote class
 RemotingConfiguration.RegisterWellKnownClientType(typeof(eLibraryServer.DAL),"tcp://localhost:8089/eLibraryServer.rem");
}
```

Step 3

- To make sure that after changes the application is calling the remote service, stop 1eLibraryHost service and start the client application.
  Since the remote service is not active, you may receive the following message:



- Start the service **1eLibraryHost** from Visual Studio panel "Server Explorer".
  Run the client application. Now the connection should be accomplished successfully. Upon activation of the ***show books***, the display looks the same as using a local connection with eLibraryServer.dll, only that we use remote connection through an established service.