

```

using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;
using System.Runtime.Remoting;
using System.Runtime.Remoting.Channels;
using System.Runtime.Remoting.Channels.Tcp;
using sb54_CSAI.remoteInterfaces;

namespace sb54_CSAI.ClientApp
{
    #region ClientRemoting CLASS
    /// <summary>
    /// Provides a new <see cref="sb54_CSAI.remoteInterfaces.IClientRemoting">
    IClientRemoting </see>
    /// object that the ReMP3 client may use to control the Remote <see cref="sb54_CSAI.
    ServerApp.FrmMain">ReMP3 Server Application </see>
    /// </summary>
    public class ClientRemoting
    {
        #region Instance Fields
        //Instance Fields
        private static IClientRemoting oRemSerrive = null;
        #endregion
        #region Public Methods
        /// <summary>
        /// Attempts to create a new <see cref="sb54_CSAI.remoteInterfaces.RemotingObject">
        RemotingObject </see>
        /// (The Remotable object), and calls its RemoteLoaderCall() method, to test the
        Remoting connection.
        /// </summary>
        /// <returns>If the RemoteLoaderCall() method calls works, True is returned, else
        false is returned to
        /// indicate the Remotable object failed.</returns>
        public static bool setupRemoting()
        {
            try
            {
                //Get a TCP channel,register the channel
                ChannelServices.RegisterChannel(new TcpChannel());
                //setup connection
                String remoteClassConnection = "tcp://" + CliConfigInfo.svrAdress + ":8000/
                TcpService";
                //setup up place holder for this type
                WellKnownClientTypeEntry remotetype = new WellKnownClientTypeEntry(typeof
                (RemotingObject),remoteClassConnection);
                //register the Remoteable object
                RemotingConfiguration.RegisterWellKnownClientType(remotetype);
                //If RegisterWellKnownClientType succeeded (NO Exception raised), then
                //set the internal field to be new RemotingServiceClass
                oRemSerrive = new RemotingObject();
                //call dummy channel testing method
                oRemSerrive.RemoteLoaderCall();
                //all ok, so retrun PASS status
                return true;
            }
            catch (Exception ex)
            {
                //something went wrong
                MessageBox.Show("Problem with Remoting in ClientRemoting : " +
                "\r\n\r\n\r\n" + ex.Message +
                "\r\n\r\n\r\n" +
                "The Re-MP3 Server application must be running in order to use\n" +
                "the Re-MP3 Client, please retry when the Re-MP3 Server application\n" +
                "has been started",
                "Remoting Services ERROR", MessageBoxButtons.OK,
                MessageBoxIcon.Error);
                //so retrun FAIL status
                return false;
            }
        }
    }
}

```

```
    }  
  
    }  
  
    /// <summary>  
    /// An IRemInterface interface that the <see cref="sb54_CSAI.ClientApp.FrmClient">  
    /// ReMP3 Client </see>will use for its Remote control operations  
    /// </summary>  
    /// <returns>An IRemInterface interface that the <see cref="sb54_CSAI.ClientApp.  
FrmClient">  
    /// ReMP3 Client </see>will use for its Remote control operations</returns>  
    public static IClientRemoting getClientInterface()  
    {  
        return oRemServive;  
    }  
    #endregion  
}  
#endregion  
}
```