**Synchronous Server Socket Example**

**.NET Framework 4**

The following example program creates a server that receives connection requests from clients. The server is built with a synchronous socket, so execution of the server application is suspended while it waits for a connection from a client. The application receives a string from the client, displays the string on the console, and then echoes the string back to the client. The string from the client must contain the string "<EOF>" to signal the end of the message.

**C#**

using System;

using System.Net;

using System.Net.Sockets;

using System.Text;

public class SynchronousSocketListener {

// Incoming data from the client.

public static string data = null;

public static void StartListening() {

// Data buffer for incoming data.

byte[] bytes = new Byte[1024];

// Establish the local endpoint for the socket.

// Dns.GetHostName returns the name of the

// host running the application.

IPHostEntry ipHostInfo = Dns.Resolve(Dns.GetHostName());

IPAddress ipAddress = ipHostInfo.AddressList[0];

IPEndPoint localEndPoint = new IPEndPoint(ipAddress, 11000);

// Create a TCP/IP socket.

Socket listener = new Socket(AddressFamily.InterNetwork,

SocketType.Stream, ProtocolType.Tcp );

// Bind the socket to the local endpoint and

// listen for incoming connections.

try {

listener.Bind(localEndPoint);

listener.Listen(10);

// Start listening for connections.

while (true) {

Console.WriteLine("Waiting for a connection...");

// Program is suspended while waiting for an incoming connection.

Socket handler = listener.Accept();

data = null;

// An incoming connection needs to be processed.

while (true) {

bytes = new byte[1024];

int bytesRec = handler.Receive(bytes);

data += Encoding.ASCII.GetString(bytes,0,bytesRec);

if (data.IndexOf("<EOF>") > -1) {

break;

}

}

// Show the data on the console.

Console.WriteLine( "Text received : {0}", data);

// Echo the data back to the client.

byte[] msg = Encoding.ASCII.GetBytes(data);

handler.Send(msg);

handler.Shutdown(SocketShutdown.Both);

handler.Close();

}

} catch (Exception e) {

Console.WriteLine(e.ToString());

}

Console.WriteLine("\nPress ENTER to continue...");

Console.Read();

}

public static int Main(String[] args) {

StartListening();

return 0;

}

}