**Using a Synchronous Client Socket**

**.NET Framework 4**

A synchronous client socket suspends the application program while the network operation completes. Synchronous sockets are not suitable for applications that make heavy use of the network for their operation, but they can enable simple access to network services for other applications.

To send data, pass a byte array to one of the [Socket](http://msdn.microsoft.com/en-us/library/system.net.sockets.socket.aspx) class's send-data methods ([Send](http://msdn.microsoft.com/en-us/library/system.net.sockets.socket.send.aspx) and [SendTo](http://msdn.microsoft.com/en-us/library/system.net.sockets.socket.sendto.aspx)). The following example encodes a string into a byte array buffer using the [Encoding.ASCII](http://msdn.microsoft.com/en-us/library/system.text.encoding.ascii.aspx) property and then transmits the buffer to the network device using the **Send** method. The **Send** method returns the number of bytes sent to the network device.

**C#**

byte[] msg = System.Text.Encoding.ASCII.GetBytes("This is a test");

int bytesSent = s.Send(msg);

The **Send** method removes the bytes from the buffer and queues them with the network interface to be sent to the network device. The network interface might not send the data immediately, but it will send it eventually, as long as the connection is closed normally with the [Shutdown](http://msdn.microsoft.com/en-us/library/system.net.sockets.socket.shutdown.aspx) method.

To receive data from a network device, pass a buffer to one of the **Socket** class's receive-data methods ([Receive](http://msdn.microsoft.com/en-us/library/system.net.sockets.socket.receive.aspx) and [ReceiveFrom](http://msdn.microsoft.com/en-us/library/system.net.sockets.socket.receivefrom.aspx)). Synchronous sockets will suspend the application until bytes are received from the network or until the socket is closed. The following example receives data from the network and then displays it on the console. The example assumes that the data coming from the network is ASCII-encoded text. The **Receive** method returns the number of bytes received from the network.

**C#**

byte[] bytes = new byte[1024];

int bytesRec = s.Receive(bytes);

Console.WriteLine("Echoed text = {0}",

System.Text.Encoding.ASCII.GetString(bytes, 0, bytesRec));

When the socket is no longer needed, you need to release it by calling the [Shutdown](http://msdn.microsoft.com/en-us/library/system.net.sockets.socket.shutdown.aspx) method and then calling the **Close** method. The following example releases a **Socket**. The [SocketShutdown](http://msdn.microsoft.com/en-us/library/system.net.sockets.socketshutdown.aspx) enumeration defines constants that indicate whether the socket should be closed for sending, for receiving, or for both.

**C#**

s.Shutdown(SocketShutdown.Both);

s.Close();