

```
Imports System.Net
Imports System.Net.Sockets
Imports System.Text

Module Module1

    Const COOKIE_PORT As Short = 1831

    Sub ExitUsage()
        Console.WriteLine("usage: Lab2ClientVB [request [server-host [port]]]")
        Console.WriteLine("where:")
        Console.WriteLine("    request is")
        Console.WriteLine("    time    for the server date and time")
        Console.WriteLine("    random  for the a random cookie (the default)")
        Console.WriteLine("    n      for cookie #n (n is a positive number)")
    End Sub

    Sub Main(ByVal cmdArgs() As String)

        Dim reqType As Integer = 2
        Dim cookieNum As Integer = 0
        Dim serverHost As String = "127.0.0.1"
        Dim serverPort As Short = COOKIE_PORT

        If cmdArgs.Length > 3 Then
            ExitUsage()
        End If

        If cmdArgs.Length >= 1 Then

            If cmdArgs(0).Equals("time") Then
                reqType = 1
            ElseIf cmdArgs(0).Equals("random") Then
                reqType = 2
            Else
                Try
                    reqType = 3
                    cookieNum = Integer.Parse(cmdArgs(0))
                Catch ex As Exception
                    Console.WriteLine(ex.Message)
                    ExitUsage()
                End Try
            End If

            If cmdArgs.Length >= 2 Then

                serverHost = cmdArgs(1)

                If cmdArgs.Length >= 3 Then
                    Try
                        serverPort = Short.Parse(cmdArgs(2))
                    Catch ex As Exception
                        Console.WriteLine(ex.Message)
                        ExitUsage()
                    End Try
                End If
            End If

            End If

            ' Data buffer for incoming data.
            Dim bytes(1024) As Byte

            ' Connect to a remote device.

            ' Establish the remote endpoint for the socket.
            Dim ipHostInfo As IPHostEntry = Dns.GetHostEntry(serverHost)
            Dim ipAddress As IPAddress = ipHostInfo.AddressList(0)
        End Sub
    End Module
```

```
Dim remoteEP As New IPEndPoint(ipAddress, serverPort)

' Create a TCP/IP socket.
Dim sender As New Socket(AddressFamily.InterNetwork, _
    SocketType.Stream, ProtocolType.Tcp)

' Connect the socket to the remote endpoint.
sender.Connect(remoteEP)

Console.WriteLine("Socket connected to {0}", _
    sender.RemoteEndPoint.ToString())

' Encode the data string into a byte array.
Dim msg As Byte() = _
    Encoding.ASCII.GetBytes(reqType.ToString() & cookieNum.ToString() & "<EOF>")

' Send the data through the socket.
Dim bytesSent As Integer = sender.Send(msg)

' Receive the response from the remote device.
Dim bytesRec As Integer = sender.Receive(bytes)

Dim sCookie As String = _
    Encoding.ASCII.GetString(bytes, 0, bytesRec)

If reqType = 1 Then
    Console.WriteLine("Server date & time is " & sCookie)
Else

    Dim sb As StringBuilder = New StringBuilder
    Dim ch As Char

    For c As Integer = 0 To sCookie.Length() - 1

        ch = sCookie.Chars(c)

        If ch = "|" Then
            sb.Append(vbCrLf)
        Else
            sb.Append(ch)
        End If
    Next

    If reqType = 2 Then
        Console.WriteLine("Random cookie:" & vbCrLf)
    Else
        Console.WriteLine("Cookie #" & cookieNum & ":" & vbCrLf)
    End If

    Console.WriteLine(WWrap(sb.ToString, 80))
End If

' Release the socket.
sender.Shutdown(SocketShutdown.Both)
sender.Close()
End Sub

' From
' http://www.visualbasic.happycodings.com/ActiveX_Controls/code6.html

'Word Wrap

Function WWrap(ByVal StrDummy As String, ByVal StrSize As Integer) As String
    If Len(Trim(StrDummy)) < StrSize Then
        WWrap = StrDummy
        Exit Function
    End If
```

```
'
Dim StrLine1 As String
Dim LoopI As Integer
Dim IntJ As Integer
Dim Padding As String

Dim sb As StringBuilder = New StringBuilder
For p As Integer = 0 To StrSize - 1
    sb.Append(" ")
Next
Padding = sb.ToString()

LoopI = 1
StrLine1 = Mid(StrDummy, LoopI, StrSize)
WWrap = ""
While Trim(StrLine1) <> ""
    If (Mid(StrDummy, (LoopI - 1) * Len(StrLine1) + 1, 1) <> " ") Then
        IntJ = InStrRev(StrLine1, " ")
        If IntJ <> 0 Then
            StrDummy = Mid(StrDummy, 1, (LoopI - 1) * StrSize + IntJ - 1) + _
                Padding.Substring(0, StrSize - IntJ + 1) + Mid(StrDummy, (LoopI - 1) * StrSize + IntJ + 1)
        End If
        LoopI = LoopI + 1
        StrLine1 = Mid(StrDummy, (LoopI - 1) * StrSize + 1, StrSize)
        StrLine1 = StrLine1 & Padding.Substring(0, StrSize - StrLine1.Length)
    End While
WWrap = StrDummy
End Function

End Module
```