

Re: how to send 1billion characters using Winsock

Source:

<http://www.tech-archive.net/Archive/VB/microsoft.public.vb.general.discussion/2008-08/msg00059.html>

- From: "expvb" <nobody@xxxxxxx>
 - Date: Mon, 4 Aug 2008 10:27:49 -0400
-

"Vincent Mark Celino" <babytoy_5@xxxxxxxxxx> wrote in message
[news:OGj0hxT9IHA.4004@xxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:OGj0hxT9IHA.4004@xxxxxxxxxxxxxxxxxxxxxxxxxxxx)

how to send a huge size of string using a winsock control? how to do that in easy way? i read more about winsock but i cant figure out how to completely get all the 1 million characters from the sender... please help me.. my problem is to receive a huge number of characters from a sender.. please someone help me...

You could do the following:

```
Dim s As String
```

```
s = Space(1000000)  
Winsock1.SendData s
```

The recipient will get multiple DataArrival events until everything is received. Note that the above code could generate runtime error 10035, so use the function SendSocketData() that I posted in your thread with subject "Winsock Client Server Application Problem". It's likely that the first send will succeed, but the second SendData will generate error 10035, so you have to try it later. SendComplete event tells the exact time when the outstanding SendData has finished. As an alternative, you can set a flag at that event to see when SendData is finished. Example:

```
' Structure for socket information  
Public Type SocketInfoT  
DataToSend As String  
bSendDataFinished As Boolean ' Initialize to True after each Connect,  
Error event.  
End Type
```

```
' Wait for any outstanding SendData request to finish  
Do While Not SocketInfo(Index).bSendDataFinished  
DoEvents ' Or Sleep 0  
Loop  
' Send the data
```

Re: how to send 1billion characters using Winsock

```
SocketInfo(Index).bSendDataFinished = False  
Winsock1(Index).SendData s
```

At SendComplete, set the flag back to indicate that SendData has finished:

```
Private Sub Winsock1_SendComplete(Index As Integer)  
SocketInfo(Index).bSendDataFinished = True  
End Sub
```

To combine multiple DataArrival data, use a module level variable(in the declaration section of the form), or a Static variable inside DataArrival procedure. Then you can combine the text until you get everything. Since you could get multiple "packets" in one DataArrival event, you have to separate the packets somehow. The example below shows how to recombine "packets" and separate them in all fragmentation situations. You use a character as a "packet" separator, such as "|", which is Chr(124), or you could use Chr(0) with the same result. VB String data type allows nulls in the middle of a string, and Len() function would not stop at the null char. You must make sure that the data you are sending does not contain the packet separator char, here is an example:

```
DataToSend = Replace(DataToSend, "|", "")  
' Add the packet separator char  
DataToSend = DataToSend + "|"
```

Here is an example of how to recombine fragmented DataArrival events(air code):

```
Private Type WinsockDataT  
DataRxSoFar As String  
End Type
```

```
Dim w As WinsockDataT
```

```
Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)  
Dim s As String  
Dim i As Long
```

```
Winsock1.GetData s
```

```
w.DataRxSoFar = w.DataRxSoFar + s
```

```
' Look for the "packet" separator.  
i = InStr(1, w.DataRxSoFar, "|", vbBinaryCompare)  
Do While i <> 0  
' Complete packet received, extract the packet  
s = Left(w.DataRxSoFar, i - 1)  
' Remove the packet from DataRxSoFar  
w.DataRxSoFar = Right(w.DataRxSoFar, Len(w.DataRxSoFar) - i - 1)  
' Parse the packet  
ParsePacket s
```

Re: how to send 1billion characters using Winsock

```
' Search again in case we received 2 or more packets in one  
DataArrival event  
i = InStr(1, w.DataRxSoFar, "|", vbBinaryCompare)  
Loop  
End Sub
```

```
Private Sub ParsePacket(ByRef s As String)
```

```
End Sub
```

Finally, you must test your DataArrival and parser separately with test data and without winosck. Include all possible fragmentation to make sure that it functions correctly before incorporating it into the main program. This will make it easier to fix any issues quickly.