```
'
'   SimpleTCPNetwork
'


   The following four events can be seen by the program and indicates
   when specific events occur

       newTCPConnection -  when a client connects to a server
                           (contains an id that can be used to identify this
client)

       closeTCPConnection - when a client (or server) shuts down the
connection
                           (contains id that is now gone)

       tcpConnectionFailed - if client cannot connect to the remote system
                             (no connection was made, hence no remote id)

       newTCPMessage - when any connection now has new data ready for it
                           (contains an id that indicates who now has new
data)


    Event newTCPConnection(ByVal conn As Integer)
    Event closedTCPConnection(ByVal conn As Integer)
    Event tcpConnectionFailed()
    Event newTCPMessage(ByVal conn As Integer)


   status() - Allow read access to the debug message

    ReadOnly Property status() As String


   serverRunning - True/False

       Indicate if the server is running and ready for connections

    ReadOnly Property serverRunning() As Boolean



   numConnections

   Return the number of connections to other systems
   (most useful for server applications)

    ReadOnly Property numConnections() As Integer



   StartServer

       Start listening for clients to connect to a specific port
       (port is 8080 by default)
```

```
Sub StartServer(Optional ByVal port As Integer = 8080)


       StopServer

       Stop listening for new connections

Sub StopServer()



       StartClient

       Connect to a server, given it's hostname and port
       (defaults to connecting on the server on the same system)

       Technically, the system can be a client and a server at the same time


Function StartClient(Optional ByVal host As String = "localhost", _
                                  Optional ByVal port As Integer = 8080) As
Boolean



      SendMessageToAll

       Send a single string to all remote systems

Sub SendMessageToAll(ByVal str As String)


     SendMessageToAllExcept

       Send a message to all other systems except one specific one
       (say like for a repeater application)


Sub SendMessageToAllExcept(ByVal str As String, ByVal who As Integer)


       SendMessageTo

       Send a string to a specific connection (given by an id)

Sub SendMessageTo(ByVal str As String, ByVal who As Integer)



       CloseConnection(id)

       Given an id, close that specific connection


Sub CloseConnection(ByVal who As Integer)
```

```
       CloseConnection()

       If passed with no arguments, close ALL of the connections
       (name is "closeConnection" to make sense for client mode)

Sub CloseConnection()


       GetMessage(id)

       Retreive the next message in the queue for a specific connection

Function GetMessage(ByVal who As Integer) As String



       GetMessage()

       Get the next message from any client
       (note: always searches from the first client, so you could starve the
later ones)

       (this function is more for client mode than server

Function GetMessage() As String


       GetMessage(name)

       Get the next message from any client and return the name of client
       at the same time

       (note: always searches from the first client, so you could starve the
later ones)

Function GetMessage(ByRef name As String) As String


       GetMessage(str,id)

       Get the next message from any client and return the id of client
       at the same time

       (note: always searches from the first client, so you could starve the
later ones)

Sub GetMessage(ByRef str As String, ByRef who As Integer)



       Name(id)

       Return a string representing the name of the client
       (Returns an empty string if the id is invalid)

Function Name(ByVal who As Integer) As Boolean
```

```
       HasMessages(id)

       Returns true if there are incoming messages waiting to be retreived
for
       the connection with "id"

       (Returns false if id is not valid)

Function HasMessages(ByVal who As Integer) As Boolean


       HasMessages()

       Indicate if ANY connetion has a message that needs read


Function HasMessages() As Boolean



       myipaddr(#)

       Given a number, return a string representing the corresponding ip
address

Function myipaddr(ByVal entrynum) As String


       myipaddr()

       Return a string representing (one of) my ip address

       Make sure that it is an IP address (and not IPv6 or some other proto)

Function myipaddr() As String



       mydnsname

       Return a string representing the name of my system by using DNS

Function mydnsname() As String
```