

' Gambas class file

' THIS EXAMPLE SHOWS HOW TO USE THE 'Socket'
' CLASS FROM 'gb.net' COMPONENT
' (C) 2003-2004 Daniel Campos Fernández
' (danielcampos@netcourrier.com)
' modifikace ok1ufc

Private CurProtocol As Integer

Public Sub Form_Open()

CurProtocol = 0
Label2.Visible = True
txtPort.Visible = True
Label1.Text = "Host :"

End

Public Sub MySocket_Ready()

' When connection proccess has finished
' successfully, "Connected" event will raise

timTimeout.Enabled = False

If CurProtocol = 1 Then

lblMessage.Text = ("Connected to path ") & MySocket.Path

Else

lblMessage.Text = ("Connected to remote host ") & MySocket.Path

lblMessage.Text = lblMessage.Text & (" - Using local address : ") & MySocket.LocalHost & ":"

& MySocket.LocalPort

End If

Me.Enabled = True

Set_Interface(True)

End

Public Sub MySocket_Closed()

' this event will raise when foreing host
' closes the socket by any reason

lblMessage.Caption = ("Connection Closed by foreign host.")

Me.Enabled = True

Set_Interface(False)

End

Public Sub MySocket_Found()

```
*****
' this event will raise when foreing host
' name has been translated to IP
*****
```

```
lblMessage.Caption = ("Host Found. Connecting...")
```

```
End
```

```
Public Sub MySocket_Read()
```

```
*****
' When some data arrives from the remote
' part of the socket, "DataAvailable" event
' is raised
*****
```

```
Dim S As String
```

```
If MySocket.Status = Net.Connected Then
    Read #MySocket, S, Lof(MySocket)
```

```
End If
```

```
End
```

```
Public Sub MySocket_Error()
```

```
*****
' this is the function to
' handle Errors when trying
' to read or write to the socket
*****
```

```
Select Case MySocket.Status
```

```
Case Net.CannotCreateSocket
```

```
    lblMessage.Text = ("The system does not allow to create a socket")
```

```
Case Net.HostNotFound
```

```
    lblMessage.Text = ("Host not Found")
```

```
Case Net.ConnectionRefused
```

```
    lblMessage.Text = ("Unable to Connect. Connection Refused")
```

```
Case Net.CannotRead
```

```
    lblMessage.Text = ("Error Reading Data")
```

```
Case Net.CannotWrite
```

```
    lblMessage.Text = ("Error Writing Data")
```

```
End Select
```

```
Set_Interface(False)
```

```
End
```

```
Private Sub Set_Interface(bState As Boolean)
```

```
*****
' An auxiliar method to enable/disable
' controls in the formulary , when
```

```

' connection is stablished or closed
'*****

btnConnect.Enabled = Not bState
txtHost.Enabled = Not bState
txtPort.Enabled = Not bState
timTimeout.Enabled = bState

End

Public Sub btnConnect_Click()

'*****
' To connect to remote host we call to
' connectsocket method, passing Host Name
' and port as arguments
'*****

Dim RetVal As Integer

btnConnect.Enabled = False
Select Case CurProtocol
Case 0
' Stablishing a TCP connection.
' Here we use Host and
' Port properties, we could also
' do directly MySock.Connect(TextBox1.Text,VAL(TextBox2.Text))
MySocket.Host = txtHost.Text
MySocket.Port = Val(txtPort.Text)
MySocket.Connect()
Case 1
' Stablishing a Local connection.
' Here we use Path and
' Port properties, we could also
' do directly MySock.Connect(TextBox1.Text,0)
MySocket.Path = txtHost.Text
MySocket.Port = Net.Local
MySocket.Connect()
End Select
If MySocket.Status > Net.Inactive Then
If CurProtocol = 0 Then
'*****
' TCP : connection in progress...
'*****

lblMessage.Text = ("Looking up host name...")
timTimeout.Delay = 10000 ' we'll wait a maximun time of 10 seconds
timTimeout.Enabled = True
End If
End If

End

```

```
Public Sub timTimeout_Timer()
```

```
'*****
```

```
' timeout trying to connect
```

```
'*****
```

```
Me.Enabled = True
```

```
timTimeout.Enabled = False
```

```
If MySocket.Status <> Net.Connected Then
```

```
    Close MySocket
```

```
    Set_Interface(False)
```

```
    lblMessage.Text = ("Timeout trying to stablsh connection")
```

```
End If
```

```
End
```

```
Public Sub Form_Close()
```

```
'*****
```

```
' Close possible Stablshed connections
```

```
'*****
```

```
If MySocket.Status > 0 Then Close #MySocket
```

```
End
```

```
Public Sub Button1_Click()
```

```
    Zap1()
```

```
End
```

```
Public Sub Button2_Click()
```

```
    Vyp1()
```

```
nd
```

```
Public Sub Button3_Click()
```

```
    Zap2()
```

```
End
```

```
Public Sub Button4_Click()
```

```
    Vyp2()
```

```
End
```

```
Public Sub Zap1()
```

```
    If MySocket.Status = Net.Connected Then
```

```
        Write #MySocket, "Zap1", 4
```

```
    End If
```

```
End
```

```
Public Sub Zap2()
```

```
    If MySocket.Status = Net.Connected Then
```

```
        Write #MySocket, "Zap2", 4
```

```
    End If
```

End

Public Sub Vyp1()

 If MySocket.Status = Net.Connected Then

 Write #MySocket, "Vyp1", 4

 End If

End

Public Sub Vyp2()

 If MySocket.Status = Net.Connected Then

 Write #MySocket, "Vyp2", 4

 End If

End