

A decorative graphic consisting of a thin yellow circle on the left side. A thick black bracket is positioned vertically on the left, overlapping the circle. A thick yellow bracket is positioned vertically on the right, also overlapping the circle. A horizontal olive-green bar with a white gradient on the right side is centered across the middle of the circle.

Windows systems Programming
Windows Sockets

Marios Papas
Neophytos Eliades

Historical Background

- Proposed by Martin Hall (JSB Software) in October 1991
- Platform
 - Windows family OS
- First specification edition by
 - Martin Hall, Mark Towfiq (Microdyne)
 - Geoff Arnold (Sun Microsystems)
 - Henry Sanders (Microsoft)
- Version 1.0 (June 1992)
 - Defined the basic operation of Winsock
 - Only TCP/UDP mentioned

[Historical Background]

■ Alternatives

- Anonymous pipes and named pipes
- Common Object Request Broker Architecture (CORBA)
- Distributed Computing Environment (DCE)
- Message Bus (MBUS) (specified in RFC 3259)
- ONC RPC
- XML XML-RPC or SOAP
- ZeroC's Internet Communications Engine (ICE)

[Why use Winsock]

- Windows Inter-process communication
 - Remotely
 - Locally
- Porting of code already written for Berkeley Sockets is straightforward.
- Windows systems easily integrate into TCP/IP networks.
- Works for both IPv4 and the emerging IPv6.
- Sockets can be used with Windows overlapped I/O
 - Allow scalability on large number of active clients
- Sockets can be treated as file handles
 - Like UNIX allows sockets used as file descriptors

[Why use Winsock]

- Supports Datagrams (UDP)
 - Connectionless
 - Unreliable small messages
- Supports transport protocol access
 - Internet Control Message Protocol (ICMP)
 - Internet Group Management Protocol (IGMP)

Why use Winsock

- Supports Reliably Delivered Messages
 - Multiple peers
 - Connectionless
 - Reliably message delivery (loss notification)
- Supports Reliable sequenced packet service
 - Two peers
 - Connection oriented
 - Reliable message delivery
- Supports Reliable stream-oriented service (TCP)
 - Two peers
 - Connection based
 - Reliable bytes stream delivery

[Installation]

- Winsock library standard with windows
 - System.Dll
- How To use winsock – Socket Class
 - Include like any other class*
 - Imports System.Net.Sockets (VB2005)
 - using System.Net.Sockets; (C#)
 - using namespace System::Net::Sockets; (C++)
 - import System.Net.Sockets.*; (Java)
- No individual Compilation needed

[Hello World! (Sender)]

```
Private Sub SendHello(ByVal server As String, ByVal port As Integer)
    Dim s As Socket = Nothing
    Dim hostEntry As IPEndPoint = Nothing
    ' Get host related information.
    hostEntry = Dns.GetHostEntry(server)
    ' Loop through the AddressList to obtain the supported AddressFamily.
    ' Avoid unexpected IP address(not belonging in the family) (Like IPv6)
    Dim address As IPAddress
    For Each address In hostEntry.AddressList
        Dim endPoint As New IPEndPoint(address, port)
        Dim tempSocket As New Socket(endPoint.AddressFamily,
        SocketType.Stream, ProtocolType.Tcp)
        Try
            tempSocket.Connect(endPoint)
        Catch ex As Exception
            MsgBox("Error Connecting to " + address.ToString)
        End Try
        If tempSocket.Connected Then
            s = tempSocket
            Exit For
        End If
    Next address
```


[Hello World! (Sender)]

```
'If s is not null
  If Not s Is Nothing Then
'Set up variables and String to write to the server.
  Dim ascii As System.Text.Encoding =
System.Text.Encoding.ASCII
  Dim msg As String = "Hello World"
  Dim memstream As New MemoryStream(200)
  Dim bw As New BinaryWriter(memstream)
  bw.Write(msg)
  Dim bytesSent As [Byte]() = memstream.ToArray
' Send request to the server.
  s.Send(bytesSent, bytesSent.Length,
SocketFlags.None)
  End If
End Sub
```

[Hello World! (Receiver)]

```
Private Sub getHello(ByVal port As Integer)
    Try
        listenSocket = New Socket(AddressFamily.InterNetwork,
            SocketType.Stream, ProtocolType.Tcp)
        ' bind the listening socket to the port
        Dim hostIP As IPAddress = IPAddress.Any
        Dim ep As New IPEndPoint(hostIP, port)
        listenSocket.Bind(ep)
        ' start listening (max number of listening sockets 10)
        listenSocket.Listen(10)
        While (True)
            dataSocket = listenSocket.Accept()
            ' The following will block until the page is transmitted.
            Dim bytesReceived() As Byte
            Dim bytes As Integer = 0
            Dim msg As String = ""
            Do
                bytes = dataSocket.Receive(bytesReceived, bytesReceived.Length, 0)
                msg = msg + System.Text.Encoding.ASCII.GetString(bytesReceived, 0, bytes)
            Loop While bytes > 0
            MsgBox("Received:" + vbNewLine + msg)
        End While
    Catch ex As Exception
        MsgBox(ex.ToString)
    End Try
End Sub
```

[Socket Class Constructor]

- **Public Sub New** (*addressFamily* As AddressFamily,
socketType As SocketType,
protocolType As ProtocolType)
- AddressFamily types:
 - AppleTalk AppleTalk address.
 - InterNetwork Address for IP version 4
 - InterNetworkV6 Address for IP version 6
- SocketType
 - Dgram (UDP)
 - Raw (ICMP)
 - Rdm (Broadcast)
 - Seqpacket (Reliable Connection Oriented)
 - Stream Supports (Reliable Connection Oriented)

[Socket Class Constructor]

- ProtocolType

- Icmp Internet Control Message Protocol.
- IcmpV6 Internet Control Message Protocol for IPv6.
- IP Internet Protocol.
- IPv4 Internet Protocol version 4.
- IPv6 Internet Protocol version 6 (IPv6).
- Tcp Transmission Control Protocol.
- Udp User Datagram Protocol.

[Socket Class Common Methods]

- Public Sub **Bind** (*localEP* As EndPoint)
 - Associates a Socket with a local endpoint.
- Public Sub **Listen** (*backlog* As Integer)
 - Places a Socket in a listening state.
- Public Function **Accept** As Socket
 - Creates a new Socket for a newly created connection.

[Socket Class Common Methods]

- Public Function **Send** (*buffer* As Byte(),
size As Integer,
socketFlags As SocketFlags)
As Integer
 - Sends the specified number of bytes of data to a connected Socket
- Public Function **Receive** (*buffer* As Byte(),
socketFlags As SocketFlags)
As Integer
 - Receives data from a bound Socket into a receive buffer

[Send – Receive Ping]

```
Private Sub SendPing(ByVal server As String)
' Get host related information.
    Dim hostEntry As IPHostEntry = Dns.GetHostEntry(server)
    Dim localhost As IPHostEntry = System.Net.Dns.GetHostByName
        (System.Net.Dns.GetHostName())

    Dim ipepServer As System.Net.IPEndPoint = New
        System.Net.IPEndPoint(hostEntry.AddressList(0), 0)
    Dim epServer As System.Net.EndPoint = CType(ipepServer,
        System.Net.EndPoint)

    Dim epFrom As System.Net.EndPoint = New
        System.Net.IPEndPoint(localhost.AddressList(0), 0)
    Dim replyBuffer(255) As Byte
' Set up variables and String to write to the server.
    Dim bytesSent As [Byte]() = makePingPacket()
    Dim tempSocket As New Socket(AddressFamily.InterNetwork,
        SocketType.Raw, ProtocolType.Icmp)
```

[Send – Receive Ping]

```
If Not tempSocket Is Nothing Then
    Try
' Send ping
        tempSocket.SendTo(bytesSent, 0, bytesSent.Length,
                           SocketFlags.None, ipepServer)
' Receive ping
        tempSocket.ReceiveFrom(replyBuffer, SocketFlags.None,
                                epServer)

        If replyBuffer(20) = 0 Then
            tslSend.Text = "ICMP Ping Successfully sent to" +
                            txtRemoteHost.Text + ":" + txtRemotePort.Text
        Else
            tslSend.Text = "Error Sending ICMP PING to " +
                            txtRemoteHost.Text + ":" + txtRemotePort.Text
        End If
    Catch ex As Exception
        MsgBox("Error you must be administrator to have
                priviledges to send a ping")
    End Try
End If
End Sub
```


[UDP Send]

```
If Not tempSocket Is Nothing Then
    Try
' Send ping
        tempSocket.SendTo(bytesSent, 0, bytesSent.Length,
                           SocketFlags.None, ipServer)
' Receive ping
        tempSocket.ReceiveFrom(replyBuffer, SocketFlags.None,
                                epServer)
' Check if ping succeeded or not
        If replyBuffer(20) = 0 Then
            tslSend.Text = "ICMP Ping Successfully sent to" +
                           txtRemoteHost.Text + ":" + txtRemotePort.Text
        Else
            tslSend.Text = "Error Sending ICMP PING to " +
                           txtRemoteHost.Text + ":" + txtRemotePort.Text
        End If
        Catch ex As Exception
            MsgBox("Error you must be administrator to have
                   priviledges to send a ping")
        End Try
    End If
End Sub
```

[UDP Send]

```
Private Sub SendUDP(ByVal server As String, ByVal port As Integer)
    Dim s As Socket = Nothing
    Dim hostEntry As IPHostEntry = Nothing
    ' Get host related information.
    hostEntry = Dns.GetHostEntry(server)
    ' Loop through the AddressList to obtain the supported AddressFamily.
    ' Avoid unexpected IP address(not belonging in the family) (Like IPv6)
    Dim address As IPAddress
    For Each address In hostEntry.AddressList
        Dim endPoint As New IPEndPoint(address, port)
        Dim tempSocket As New Socket(endPoint.AddressFamily,
                                     SocketType.Dgram, ProtocolType.Udp)
        tempSocket.Connect(endPoint)
        If tempSocket.Connected Then
            s = tempSocket
            Exit For
        End If
    Next address
```

[UDP Send]

```
If Not s Is Nothing Then
' Set up variables and String to write to the server.
  Dim i As Integer
  pbReceive.Style = ProgressBarStyle.Blocks
  For i = 0 To 1000
    Dim memstream As New MemoryStream(10)
    Dim bw As New BinaryWriter(memstream)
    bw.Write(i)
    Dim bytesSent As [Byte]() = memstream.ToArray
' Send request to the server.
    s.Send(bytesSent, bytesSent.Length,
           SocketFlags.None)
    tslSend.Text = "Sent Packet with Data " +
i.ToString
    pbSend.Value = i
  Next i
End If
End Sub
```

[UDP Receive]

```
Private Sub handleConnectionUDP()  
    Try  
        'Retrieve data from socket  
        Dim inc(200) As Byte  
        Dim msg As Integer  
        Dim i = 0  
        While i < 1000  
            dataSocket.Receive(inc, 200, SocketFlags.None)  
            Dim memstream As New MemoryStream(inc)  
            Dim br As New BinaryReader(memstream)  
            msg = br.ReadInt32  
            tslReceive.Text = "Received Packet " + msg.ToString  
            pbReceive.Value = msg  
        End While  
    Catch ex As Exception  
        MsgBox(ex.ToString)  
    End  
    Finally  
        tslReceive.Text = "Listening for incoming connections ..."  
    End Try  
End Sub
```

[UDP Receive]

```
Private Sub acceptConnections()  
    Try  
        Dim port As Integer = Convert.ToInt32(txtLocalPort.Text)  
        listenSocket = New Socket(AddressFamily.InterNetwork, SocketType.Stream,  
                                   ProtocolType.Tcp)  
  
        ' bind the listening socket to the port  
        Dim hostIP As IPAddress = IPAddress.Any  
        Dim ep As New IPEndPoint(hostIP, port)  
        listenSocket.Bind(ep)  
  
        ' start listening  
        listenSocket.Listen(10)  
        While (True)  
            dataSocket = listenSocket.Accept()  
            Dim child As New Threading.Thread(AddressOf handleConnection)  
            child.Start()  
  
        End While  
        Catch ex1 As Threading.ThreadAbortException  
  
        ' empty  
        Catch ex2 As Exception  
            MsgBox(ex2.ToString)  
  
        End Try  
    End Sub
```

[Conclusions]

- Winsock control provides relatively straightforward “Internet” programming.
- Winsock is the foundation of Internet capability in Microsoft’s OS, allowing browsers, FTP, chat and terminal applications
- Strings can be sent between client and server through a chat program.
- Can upload data to a web server and view “real-time” data acquisition with browsers!

[Bibliography]

- <http://www.wikipedia.org>
- <http://msdn.microsoft.com/>
- http://en.wikipedia.org/wiki/Berkeley_sockets
- <http://tangentsoft.net/wskfaq/>
- <http://www.sockets.com/>

[Questions

]

?