

Forum: [Hack Forums](#)

Thread: [NamedPipes](#)

KingDeath360 - January 4, 2015, 5:39 pm

alternative to sockets, networking library

I haven't seen any posts regarding this except for nanocore.

When i was looking into pipes it led me to Net.Remoting and it turns out that is "not recommended " so i was browsing the web and saw this source code and might as well post it on hf instead of having to split you data with the first byte as the id this will handle all that stuff This is an example and its not user friendly so you will have a hard time leaching it.

example uses are:

[+]Chat system

[+]Plugin System

[+]RAT

Ok so heres how to use it

\\127.0.0.1\pipe\generalstuffPipe1

127.0.0.1 ip address or just one dot for the same computer

generalstuffPipe1 pipename

Code Here is a typical loop back file transfer on our test system:

TCP/IP Transfer time: 2.5 Seconds

Named Pipes Transfer time: 3.1 Seconds

Now, if you go outside the machine and connect to a remote computer on your network the performance for named pipes is much worse:

TCP/IP Transfer time: 12 Seconds

Named Pipes Transfer time: 2.5 Minutes (Yes Minutes!)

<http://stackoverflow.com/questions/10872557/how-slow-are-tcp-sockets-compared-to-named-pipes-on-windows-for-localhost-ipc>

isn't mine

[Download](#)

[Virus Scan](#)

Hex Hooves - January 4, 2015, 6:29 pm

What the hell are you even talking about.

(Jan 4, 2015 06:39 PM)KingDeath360 Wrote: [\[CITE\]](#) [+]i think you know this one I don't actually know what you are referring to.

(Jan 4, 2015 06:39 PM)KingDeath360 Wrote: [\[CITE\]](#) Ok so heres how to use it

\\127.0.0.1\pipe\generalstuffPipe1

127.0.0.1 ip address or just one dot for the same computer

generalstuffPipe1 pipenameWhat kind of example is this? There is zero context.

GenericSkid - January 4, 2015, 6:44 pm

(Jan 4, 2015 07:29 PM)Hex Hooves Wrote: [\[CITE\]](#) What kind of example is this? There is zero

context.

Here's some context!

Code Public Class Form1

Dim WithEvents server As New NamedPipes.PipeListener("generalstuffPipe1", 1024)

Dim CID As Integer

Dim SID As Integer

Private mServerClients As New List(Of IO.Stream)

Private myClients As New List(Of NamedPipes.PipeClient)

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)

Handles MyBase.Load

End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

Handles Button1.Click

Try

server.StopListenning()

WriteLine("Server stopped listenning - client connections are still valid")

Catch ex As Exception

WriteLine(ex.Message)

End Try

End Sub

Private Sub P_ClientConnected(ByVal sender As Object, ByVal e As
NamedPipes.ClientConnectedEventArgs) Handles server.ClientConnected

Dim Handler As New iHandleClient(AddressOf HandleClient)

Handler.BeginInvoke(e.ClientStream, Nothing, Nothing)

End Sub

Private Delegate Sub iWriteLine(ByVal line As String)

Private Sub WriteLine(ByVal line As String)

Me.ListBox1.Items.Add(line)

End Sub

Private Delegate Sub iHandleClient(ByVal ClientStream As IO.Stream)

Private Sub HandleClient(ByVal ClientStream As IO.Stream)

Dim myID As Integer = SID

SID += 1

Dim WL As New iWriteLine(AddressOf WriteLine)

Dim BytesRead As Integer

Dim Buffer(1023) As Byte

mServerClients.Add(ClientStream)

Me.BeginInvoke(WL, "Server thread " & myID & ": Start")

While True

Do

BytesRead = ClientStream.Read(Buffer, 0, Buffer.Length)

If BytesRead <> 0 Then

Dim incommingMessage As String = System.Text.Encoding.Unicode.GetString(Buffer, 0,
BytesRead)

Me.BeginInvoke(WL, "Client -> Server " & myID & ": " & incommingMessage)

Dim outBuffer As Byte() = System.Text.Encoding.Unicode.GetBytes("Response to [" &
incommingMessage & "])

ClientStream.Write(outBuffer, 0, outBuffer.Length)

Erase outBuffer

```

        End If
    Loop Until BytesRead = 0
End While

' mServerClients.Remove(ClientStream)
' ClientStream.Close()
' Me.BeginInvoke(WL, "Server thread " & myID & ": Stop")
End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
    Dim D As New iStartClient(AddressOf StartClient)
    D.BeginInvoke(Nothing, Nothing)
End Sub

Private Delegate Sub iStartClient()
Sub StartClient()
    Dim myID As Integer = CID
    CID += 1
    Dim C As New NamedPipes.PipeClient("\\.\pipe\generalstuffPipe1")
    Dim WL As New iWriteLine(AddressOf WriteLine)
    Dim Hail As Byte() = System.Text.Encoding.Unicode.GetBytes("Hello from client " & myID)
    C.Timeout = 1000
    Try
        C.Connect()
        Me.BeginInvoke(WL, "Client " & myID & ": Connected")
    Catch ex As Exception
        Me.BeginInvoke(WL, "Client " & myID & ": " & ex.Message)
    Exit Sub
    End Try
    myClients.Add(C)
    Dim Buffer(1023) As Byte
    Dim BytesRead As Integer
    Dim ClientStream As IO.Stream = C.ClientStream
    ClientStream.Write(Hail, 0, Hail.Length)
    Do
        BytesRead = ClientStream.Read(Buffer, 0, Buffer.Length)
        If BytesRead <> 0 Then
            Dim incommingMessage As String = System.Text.Encoding.Unicode.GetString(Buffer, 0,
BytesRead)
            Me.BeginInvoke(WL, "Server -> Client " & myID & ": " & incommingMessage)
        End If
    Loop Until BytesRead = 0
    ' C.Close()
    ' myClients.Remove(C)
    ' Me.BeginInvoke(WL, "Client " & myID & ": Stop")
End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button3.Click
    Try
        server.StartListenning()
        WriteLine("Server started listenning")
    Catch ex As Exception
        WriteLine(ex.Message)
    End Try
End Sub

```

```
End Try
End Sub
```

```
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button4.Click
    Dim outBuffer As Byte() = System.Text.Encoding.Unicode.GetBytes(TextBox1.Text)
    For Each client As IO.Stream In mServerClients
        client.Write(outBuffer, 0, outBuffer.Length)
        client.Flush()
    Next
    Erase outBuffer
End Sub
```

```
Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button5.Click
    For Each client As IO.Stream In mServerClients
        client.Close()
    Next
End Sub
```

```
Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button6.Click
    For Each client As NamedPipes.PipeClient In myClients
        client.Close()
    Next
End Sub
```

```
End Class
```

Just C/P'd from the linked zip file.

ubbelol - January 5, 2015, 1:45 am

(Jan 4, 2015 06:39 PM)KingDeath360 Wrote: [\[CITE\]](#) I haven't seen any posts regarding this except for nanocore.

I don't think nanocore uses named pipes? From what I know it's all sockets. Also, keep in mind using pipes instead of a TCP/IP socket will most likely negatively impact performance as they're created for different purposes.