TNMEcho component

Heirarchy Properties Methods Events Tasks

Unit

NMEcho

Description

The TNMEcho component is used for sending text to an internet echo server, and having that text echoed back to you, as described in RFC 862.

TNMEcho Properties

TNMEcho
Legend

In TNMEcho

🔑
▷
▶ ElapsedTime

Derived from TPowersock

🟩 About
▷ BeenCanceled
▷
▶ BeenTimedOut
▷
▶ BytesRecvd
▷
▶ BytesSent
▷
▶ BytesTotal
🔑
▷
▶ Connected
▷
▶ Handle
🔑
🟩 Host
▷ LastErrorNo
▷
▶ LocalIP
🔑
🟩 Port
🟨 Proxy
🟨 ProxyPort
▷
▶ RemoteIP
▷
▶ ReplyNumber
🟩 ReportLevel
🔑
▷
▶ Status
🟩 TimeOut
▷
▶ TransactionReply
▷
▶ WSAInfo

Derived from TComponent

▶ ComObject
▶ ComponentCount
ComponentIndex
▶ Components
▶ ComponentState
▶ ComponentStyle
DesignInfo

🟩 Tag	▶ Owner
	VCLComObject

TNMEcho Methods

TNMEcho
Legend

In TNMEcho

- 🔑 Echo

Derived from TPowersock

- Abort
- 🔑 Accept
- Cancel
- 🔑 CaptureFile
- 🔑 CaptureStream
- 🔑 CaptureString
- CertifyConnect
- 🔑 Connect
- Create
- Destroy
- 🔑 Disconnect
- FilterHeader
- GetLocalAddress
- GetPortstring
- 🔑 Listen
- 🔑 read
- 🔑 ReadLn
- RequestCloseSocket
- SendBuffer
- 🔑 SendFile
- 🔑 SendStream
- 🔑 Transaction
- 🔑 write
- 🔑 writeln

Derived from TComponent

- DestroyComponents
- Destroying
- FindComponent
- FreeNotification
- FreeOnRelease
- GetParentComponent
- HasParent
- InsertComponent
- RemoveComponent
- SafeCallException

Derived from TPersistent

- Assign
- GetNamePath

Derived from TObject

- ClassInfo
- ClassName
- ClassNameIs
- ClassParent
- ClassType
- CleanupInstance

DefaultHandler
Dispatch
FieldAddress
Free
FreeInstance
GetInterface
GetInterfaceEntry
GetInterfaceTable
InheritsFrom
InitInstance
InstanceSize
MethodAddress
MethodName
NewInstance

TNMEcho Events

TNMEcho
Legend

Derived from TPowersock

- 🟨 OnAccept
- 🔑 OnConnect
- OnConnectionFailed
- 🟨 OnConnectionRequired
- 🔑 OnDisconnect
- 🟨 OnError
- OnHostResolved
- OnInvalidHost
- OnPacketRecvd
- OnPacketSent
- OnRead
- OnStatus

About the TNMEcho component

TNMEcho reference

Purpose

The purpose of the TNMEcho component is to send text to the remote echo host, and having that text echoed back to you. This is used most often for testing network integrity and speed.

RFC: RFC 862

Tasks

Before sending text to the server, a connection must be established. Establishing a connection is done by setting the Host property to a valid internet echo server, and the Port property to the corresponding port. Please note that Echo servers usually listen on well-known port 7.

Echoing text to the server:

Sending text to the server is done with the Echo method. The string passed as the parameter to this method will be the return value once the server sends the text back.

ElapsedTime property

See also Example

Declaration

```
property ElapsedTime: single;
```

Description

The ElapsedTime property is the time it takes for the server to echo the text sent with the Echo method back to the client

Scope: Public
Accessibility: RunTime, ReadOnly

Notes:
This property won't be set unless an Echo is called, and will contain the result from the last echo until another echo is complete successfully.

See also

Echo method

Example

To recreate this example, you will need to create a new blank Delphi application.

Place a TEdit, 2 TLabels, 3 TButtons, and a TNMEcho on the form.

Insert the following code into Button1's OnClick event:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  NMEcho1.Host := 'www.netmastersllc.com';
  NMEcho1.Connect;
end;
```

Insert the following code into Button2's OnClick event:

```
procedure TForm1.Button2Click(Sender: TObject);
begin
  NMEcho1.Disconnect;
end;
```

Insert the following code into Button3's OnClick event:

```
procedure TForm1.Button3Click(Sender: TObject);
begin
  Label1.Caption := NMEcho1.Echo(Edit1.Text);
  Label2.Caption := FloatToStr(NMEcho1.ElapsedTime);
end;
```

Example Description:
When this application is run, click Button1 to connect to the remote host, which is set to www.netmastersllc.com. Button2 is used to disconnect the client from the remote host when echoing has been finished. Button3 echoes the text in Edit1, and whatever text is returned is displayed in Label1's caption. Label2's caption is set to the ElapsedTime property (a string representation thereof), which displays how many milliseconds it took for the data to get echoed.

Echo method

See also Example

Declaration

```
function Echo(EchoString: string): string;
```

Description

The Echo method sends the text passed in EchoString to the remote server. It returns the text that is echoed back, which should be the same as the EchoString parameter.

Parameters:
The EchoString parameter is the text to be echoed to the server.

Return Value:
The text echoed back is returned as a string. It should be the same as the text sent in the EchoString parameter, with the exception of a Carrieage return/Line feed appened to it.

Notes:
You must be Connected to the remote host before calling this method. If a connection is not present, the OnConnectionRequired event is called.

See also

ElapsedTime property

Legend

- ▶ Run-time only
- ▶ Read-Only
- 🟩 Published
- 🟨 Protected
- 🔑 Key item

Heirarchy

TObject
|
TPersistent
|
TComponent
|
TPowersock