Getting Started with the CITelnet Form

Related Topics

This topic introduces the Telnet protocol and its implementation as the Crescent Internet ToolPak Telnet Form (CITelnet). You can access these topics:

- What Telnet is
- What the CITelnet form is
- Using the CITelnet form

The last topic, Using the CITelnet form, describes how to set up an application to support Telnet.

What is Telnet?

Related Topics

The Telnet protocol supports remote login over the Internet. It allows users to login to a Telnet host to execute commands and to access services as if the user were locally connected.

What is the CITelnet Form?

Related Topics

The CITELNET form enables developers to add Telnet support to a Visual Basic application. It includes these components:

- A Visual Basic form

  The CITelnet form, though not displayed, contains the CITCP custom control. The CITCP custom control provides the functionality that enables the CITelnet form to communicate with aTelnet host.

- A telnet emulator

  The Telnet emulator is a specially licensed version of the PDQComm for Windows custom control. This custom control embeds serial communications functionality into a Visual Basic application, allowing the application to transmit and receive data through a serial port. The specially licensed version supplied with the CITelnet form allows use of emulation functions only.

You set the PDQComm and CITCP properties through the CITelnet form by simply setting CITelnet properties and calling CITelnet methods. All of the host connection information or emulation data required is managed through CITelnets interface. When you call a method, the control simply uses the data that youve supplied on the property sheet and submits the request to the Telnet host.

Using the CITelnet Form

Related Topics

This topic describes how to add Telnet support to and to remove Telnet support from a Visual Basic application.

To add Telnet support to a Visual Basic application, follow these steps:

1. Start a new project or open an existing project.

2. Add the CITCP control and the special PDQComm control to the project using the Visual Basic Tools Custom Controls menu.

3. Add the CITelnet.frm to the open project using the Visual Basic File Add File menu.

4. Add the PDQComm control on a new or existing form (not the CITELNET.FRM).

5. Identify the Telnet form to the application by adding the following code to the Declarations section of the form on which the PDQComm control resides:

```
Public MyTelnet As Form
```

6. Create the Telnet object with this code:

```
Form_Load()
Set MyTelnet = New CITELNET
MyTelnet.Create PDQComm1
```

This code also identifies which PDQComm control to associate with the CITelnet form. Once the association is made, when you set the properties for the CITelnet form, they are automatically propagated to the correct PDQComm control.

7. In the PDQComm1_Keypress() event, add this code:

```
PDQComm1_KeyPress()
MyTelnet.KeyPress KeyAscii
```

This code sends the applications keystrokes to the hidden CITCP control which communicates with the Telnet host.

Once you have completed the setup steps, simply set the desired properties (like HostName or HostAddress) and use the Connect and Disconnect methods to access the Telnet host.

To remove Telnet support from a Visual Basic application, do the following:

1. In the Form_Unload module, add this code to delete the Telnet object:

```
Form_UnLoad()
Unload MyTelnet
Set MyTelnet = Nothing
```

The Crescent Internet ToolPak Telnet Form

Related Topics

Form Module

CITelnet.FRM

Object Type

CITelnet

Purpose

The Crescent Internet ToolPak CITelnet form provides programmatic access to the Telnet protocol. The CITelnet form complies with the Telnet standards defined in RFC 854.

Properties

Connected c	Echo c	Emulation c
ErrorCode c	HostAddress c	HostName c
Service c		

Methods

Connect c	Create c	Disconnect c

c A custom or modified property or method.

Connected Property

Applies To

CITelnet

Purpose

The Connected property returns the current connection status.

Syntax

[*Form.***]***CITelnet*.Connected**[** = *state$***]**

Data Type

Boolean

Usage

Read only at runtime.

Comments

If you are currently connected, the Connected property has a value of True.

CITelnet updates the Connected property when you establish a connection (with the Connect method) or terminate a connection (with the Disconnect method).

See Also

Connect, Disconnect

Echo Property

Applies To

CITelnet

Purpose

The Echo property sets or returns the state of the Telnet echo option.

Syntax

[*Form.***]***CITelnet*.Echo**[** = *state$***]**

Data Type

Boolean

Usage

Read/Write at runtime.

Comments

Set the Echo property before calling the Connect method. When Echo is set to True, CITelnet requests that the host echo typed characters. If the host acknowledges the request, Echo remains True. If the host refuses to echo, CITelnet sets the Echo to False. Toggling this property during an active connection will cause Telnet to enable and disable host echoing.

See Also

Connect

Emulation Property

Applies To

CITelnet

Purpose

The Emulation property sets the name of the emulation used when negotiating the terminal type with the host.

Syntax

[*Form.***]***CITelnet*.Emulation**[** = *string$***]**

Data Type

String

Usage

Read/Write at runtime.

Comments

Set the emulation property before calling the Connect method. The predefined emulations are:

ansi	ANSI	DEC-VT100	vt100

The duplication is required because different hosts recognize different spellings for a given emulation. You can modify the form source code to add new terminal types as needed in the Emulation (PropertyLet) procedure.

See Also

Connect

ErrorCode Property

Applies To

CITelnet

Purpose

The ErrorCode property returns the value of any WINSOCK errors.

Syntax

```
[integer%] = [Form.]CITelnet.ErrorCode
```

Data Type

Integer

Usage

Read only at runtime.

Comments

The Telnet form updates the ErrorCode property anytime a WINSOCK error occurs.

See Also

Connect

HostAddress Property

Applies To

CITelnet

Purpose

The HostAddress property sets or returns the TCP/IP address of the host computer with which you hope to establish a Telnet connection.

Syntax

[*Form.***]***CITelnet*.HostAddress**[** = *string$***]**

Data Type

String

Usage

Read/Write at runtime.

Comments

Set HostAddress (or HostName) before calling the Connect method. If this property is blank, CITelnet updates this property with the host address upon a successful Telnet connection. If you supply both the HostAddress and the HostName property, CITelnet attempts to connect using the HostName value.

You cannot modify is property during an active connection. CITelnet clears this property at disconnect.

See Also

Connect, HostName, Disconnect

HostName Property

Applies To

CITelnet

Purpose

The HostName property sets or returns the DNS name of the host computer with which you hope to establish a Telnet connection.

Syntax

[*Form.***]***CITelnet*.HostName**[** = *string$***]**

Data Type

String

Usage

Read/Write at runtime

Comments

Set HostName (or HostAddress) before calling the Connect method. If this property is blank, CITelnet updates this property with the host name upon a successful Telnet connection. If you supply both the HostAddress and the HostName property, CITelnet attempts to connect using the HostName value.

You cannot modify this property during an active connection. CITelnet clears this property at disconnect.

See Also

Connect, HostAddress, Disconnect

Service Property

Applies To

CITelnet

Purpose

The Service property sets the TCP/IP port number or service.

Syntax

[*Form.***]***CITelnet*.Service**[** = *string$***]**

Data Type

String

Usage

Read/Write at runtime.

Comments

Set the service property before calling the Connect method. The following services are predefined:

FTP	HTTP	NNTP
POP3	SMTP	TELNET

and are set like this:

```
Telnet.Service = FTP
```

The default value is TELNET.

You can also set it directly to the port you want to use as shown:

```
Telnet.Service = 8080
```

The port numbers are defined in the Service(PropertyLet) procedure. Note that if you set the Service property to an integer, the CITelnet form does not assume any type of service.

Connect Method

Applies To

CITelnet

Purpose

The Connect method establishes a Telnet connection.

Syntax

CITelnet.Connect

Data Type

Boolean

Comments

When a session is successfully established, the method returns True. Otherwise, it returns False. You must set HostName or HostAddress property before calling Connect.

See Also

Disconnect, HostName, HostAddress

Create Method

Applies To

CITelnet

Purpose

The Create method creates an instance of the CITelnet object.

Syntax

```
CITelnet.Create(objEmulator As Object)
```

Data Type

None

Comments

The objEmulator parameter must be the name of an existing PDQComm control.

You must call the Create method after you create a new Telnet object. For example:

```
DimTelnet as CITelnet
Set Telnet = NewCITelnet
Telnet.Create CITCP1, PDQComm1
```

The Create method sets references to an existing CITCP control and a PDQComm control. Note that the CITelnet control does not care about in the control. All properties that are part of CITelnet supersede the controls properties.

Disconnect Method

Applies To

CITelnet

Purpose

The Disconnect method disconnects an active Telnet session.

Syntax

CITelnet.Disconnect

Data Type

Boolean

Comments

If Disconnect returns True, there is an active connection to disconnect. It returns a False when there is no connection.

See Also

Connect, Connected