

VT220 ActiveX Control

**VT220 Emulation
ActiveX Control
for Microsoft® Windows™**

Version 5.2

**Copyright © 1995 - 2003 by Distinct Corporation
All rights reserved**

Distinct Corporation

3315 Almaden Expressway
San Jose, CA 95118 USA

Phone: +1 408-445-3270

Fax: +1 408-445-3274

Email: sales@distinct.com

WWW: <http://www.distinct.com>

Disclaimer

Distinct Corporation makes no warranties as to the contents of this documentation and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Information in this manual is subject to change without notice and does not represent a commitment on the part of Distinct Corporation. The Software described in this manual is furnished under a license agreement and may be used or copied only in accordance with the terms of that agreement.

Copyright Notice

© 1995 - 2003 by Distinct Corporation. All rights reserved.

No part of this publication may be reproduced, transmitted or translated into any language by any means without the express written permission of Distinct Corporation.

Trademarks

Distinct is a registered trademark and Visual Internet Toolkit and Distinct VT220 are trademarks of Distinct Corporation. Windows is a registered trademark of Microsoft Corporation. Other product names are trademarks or registered trademarks of their respective owners.

Last updated June 5th, 2003

Published in the United States of America

Table of Contents

Table of Contents.....	1
1 VT220 ActiveX Overview.....	3
1.1 VT220 Introduction.....	3
1.2 Usage of the VT220 Component.....	3
1.3 VT220 Property Summary.....	5
1.4 VT220 Event Summary.....	6
1.5 VT220 Method Summary.....	7
1.6 D_VT220.TXT.....	8
2 VT220 Properties.....	11
2.1 Action.....	11
2.2 BinaryData.....	12
2.3 CaptureAppend.....	13
2.4 CaptureData.....	14
2.5 CaptureDataFile.....	15
2.6 CaptureFile.....	16
2.7 CaptureScreen.....	17
2.8 CaptureScreenFile.....	18
2.9 ClearClipboard.....	19
2.10 Column.....	20
2.11 ConfigurationName.....	21
2.12 CopyClipboard.....	22
2.13 CurrentFont.....	23
2.14 EnableClipboard.....	24
2.15 Host.....	25
2.16 InCaptureData.....	26
2.17 InCaptureDataFile.....	27
2.18 NumLockMode (Windows 95 only).....	28
2.19 PasteClipboard.....	29
2.20 Port.....	30
2.21 Row.....	31
2.22 SaveSettings.....	32
2.23 ShowHelp.....	33
2.24 ShowScrollbars.....	34
2.25 TransModeStatus.....	35
2.26 UseVariant.....	36
2.27 WriteData.....	37
2.28 VtAction.....	38
3 VT220 Events.....	41
3.1 OnClose.....	41
3.2 OnConnect.....	42
3.3 OnData.....	43
3.4 OnDataB.....	44
3.5 OnError.....	45
3.6 OnList.....	49
4 VT220 Methods.....	50
4.1 Abort.....	50
4.2 Clear.....	51
4.3 ConfigureColor.....	52
4.4 ConfigureKeyboard.....	53
4.5 ConfigureTerminal.....	54

4.6	Connect	55
4.7	Copy	56
4.8	CurrentLocation	57
4.9	CustomFont	58
4.10	DataB	59
4.11	DataCapture	60
4.12	DeleteConfig	61
4.13	Destroy	62
4.14	Disconnect	63
4.15	GetSelectedText	64
4.16	GetScreenText	65
4.17	GetText	66
4.18	Initialize	67
4.19	ListConfig	68
4.20	ModifyConfig	69
4.21	Paste	70
4.22	PrinterFont	71
4.23	SaveConfig	72
4.24	SaveTermSetting	73
4.25	SelectPrinter	74
4.26	ScreenCapture	75
4.27	Transparent	76
4.28	VtFont	77

1 VT220 ActiveX Overview

1.1 VT220 Introduction

The Distinct VT220 ActiveX control allows you to integrate VT220 emulation and optional login capabilities into your applications. It has built-in Telnet capabilities to connect to the remote server. An application with VT220 emulation allows users to remotely access applications that require a VT220 terminal. This ActiveX control supports full VT52, VT100 and VT220 emulation and includes extensive keyboard mapping capabilities. This ActiveX control takes care of establishing the TCP connection, handling option negotiations, sending user input to the server and displaying incoming data in a VT220 emulation window. As an option, the application can handle establishing a connection and sending and receiving data itself. This allows the application to use a transport protocol other than TCP/IP (such as a direct modem dialup).

The combined use of the Distinct VT220 control and the Distinct Windows Sockets control allow you to do data manipulation. An example of when you may need this is an application that reads a file containing customer account data that needs to be entered into an existing billing system's user interface. The application needs to process the file data and, through VT220 emulation, enter it through the user interface. We provide a sample with source to illustrate the combined use of the VT200 and Windows Sockets controls. This sample is available for the C++ and Visual Basic environments.

The main features of this control include the ability to:

- create and manage multiple configuration profiles each with their own settings.
- select the standard deck fonts or a custom font.
- remap the keyboard. A full keyboard remapping capability with a built in pre-programmed graphical representation of the keyboard to facilitate the remapping of keys in your application.
- capture all incoming data to file.
- capture screen data to a file.
- manipulate incoming data (when used in conjunction with the Windows Sockets control)
- cut and paste data from and to the emulation window via the Windows clipboard.

1.2 Usage of the VT220 Component

See the section entitled "Using Distinct ActiveX controls in various environments" on how to add the control to your project.

After placing a VT220 control into a form, some properties can be preset at design time. The Port property defaults to server port 23 (the default Telnet port) and may be changed at design time or at run time before a connection is established. The Host property is usually set at run time right before a session is established. This allows the application to request the host name or internet address of the remote server from the user. If an application will always connect to the same host, then the Host property can also be set at design time to minimize user interaction. If an application will handle the connection itself, then the Port and Host properties are not used.

The VT220 ActiveX control contains a VT220 emulation window, which is created when the Initialize method is called (or the Action property is set to ACTION_INITIALIZE) (usually while the parent form is loaded at run time). Once the VT220 emulation window is created, the application can choose to establish the connection itself or to allow the VT220 ActiveX control to take care of handling the connection.

To allow the VT220 ActiveX control to take care of handling the connection, the connect method is called (or the value of the Action property must be set to ACTION_CONNECT.) Before a connection can be made the Host and Port properties must be set up to specify the remote server and port to connect to. If the connection can be established, then the OnConnect event will occur before the next line of code is reached. At this point, the VT220 control will handle all emulation and will transfer data to and from the remote server. In this case, the application has no control over the data being transferred.

If an application requires more control over the data being transferred, then the application can establish its own connection directly or by using other ActiveX controls (such as the Distinct Windows Sockets ActiveX control). In this case, it is the application's responsibility to send and receive data. The application must use the WriteData property to display data it received from the remote server in the emulation window and the VT220 ActiveX control will deliver user input (key strokes) to the application through OnData events. Then, the application must send the user input to the remote server. This approach is useful if the application wants to filter or change unwanted characters from the server before they are displayed. It is also possible to change user keystrokes (for example, to expand macros) before sending them to the server.

Most of the VT_ actions can only be performed once the emulation window has been created by calling the Initialize method (or using ACTION_INITIALIZE). There are some operations such as listing, modifying, deleting and saving configuration profiles, achieved by calling the ListConfig, ModifyConfig, DeleteConfig and SaveConfig methods, which can be called before a VT session is initialized. The VtAction property allows the application to configure the terminal; set the color and font; select a printer; remap keys; perform clipboard operations; list existing configurations or to modify, add, save and delete configurations. Most of these operations can also be achieved by using the corresponding methods.

The boolean properties CopyClipboard, PasteClipboard, ClearClipboard, CaptureScreen, CaptureScreenFile, CaptureData, CaptureDataFile, InCaptureData, InCaptureDataFile, TransModeStatus and SaveSettings can be used by the application to determine whether an operation is allowed at that particular moment. The operation can only be performed if the corresponding boolean property is TRUE. The boolean properties are read-only and cannot be set at any time.

The application can create a number of session configuration profiles. The ConfigurationName property must be set to the name of the terminal configuration that the application will use for the session. The application can save the current terminal settings as a configuration and then load this configuration as the terminal settings for another session. To obtain a list of all available configurations, call the ListConfig method (or set the VtAction property to VT_LIST_CONFIGURATION.) One or more OnList events will occur to deliver the names of all defined configurations.

The CurrentFont property specifies the font currently in use. To use a different terminal font, set the CurrentFont property to a different value.

The CaptureAppend property determines whether the contents of the file specified by the CaptureFile property be overwritten or appended during a screen or data capture.

The Row and Column properties specify the current cursor location on the terminal window.

The ShowHelp property determines whether the VT220 ActiveX control should show or hide the help buttons in the dialog boxes it displays.

If the application does not handle the connection itself, then it must close the connection once it is no longer needed by calling the Disconnect method (or setting the Action property to ACTION_DISCONNECT.) After the session is disconnected, the OnClose event will occur before the next line of code is reached. An application can connect (ACTION_CONNECT or Connect method) and disconnect (ACTION_DISCONNECT or Disconnect method) a session (with the same or a different server) as often as needed. Before terminating the application, it must also destroy the VT220 emulation window by calling the Destroy method (or setting the Action property to ACTION_DESTROY).

1.3 VT220 Property Summary

Action

Initialize or destroy VT220 emulation window or connect, disconnect or abort VT220 session

BinaryData

Contains the binary data received following a call to the RetrieveB method.

CaptureAppend

Overwrite or append incoming data to CaptureFile

CaptureData

Check if possible to capture incoming data to file

CaptureDataFile

Check if possible to capture incoming data to CaptureFile

CaptureFile

File name to save the captured data.

CaptureScreen

Check if possible to capture screen to file

CaptureScreenFile

Check if possible to capture screen to CaptureFile

ClearClipboard

Check if possible to clear clipboard text

Column

Column position of the cursor

ConfigurationName

Name of configuration to be modified, saved or deleted

CopyClipboard

Check if possible to copy text to clipboard

CurrentFont

Name of font that is currently in use

EnableClipboard

Enable or disable use of mouse buttons for accessing the Windows clipboard.

Host

Name of host or dotted decimal internet address

InCaptureData

Check if data capture is enabled

InCaptureDataFile

Check if data capture to CaptureFile is enabled

NumLockMode

Set the behavior of the NumLock key on the keyboard

PasteClipboard

Check if possible to paste from clipboard

Port

Remote service port on host

Row

Row position of the cursor

SaveSettings

Check if possible to save configuration

ShowHelp

Display or hide dialog box help buttons

ShowScrollbars

Display or hide scrollbars

TransModeStatus

Transparent mode status

UseVariant

Receive binary or ascii data

WriteData

Display data in VT220 emulation window

VtAction

Set up terminal, color, font, keyboard or printer and list, save, modify or delete configurations

1.4 VT220 Event Summary

OnClose

Remote connection has been closed

OnConnect

Remote connection has been established

OnData

Data from VT220 emulation window to be sent to remote server

OnDataB

Data from VT220 emulation window to be sent to remote server as binary

OnError

Local error has occurred

OnList

Listing of configuration names

1.5 VT220 Method Summary

Abort

Abort VT220 session

Clear

Clear text in clipboard

ConfigureColor

Set background and foreground colors

ConfigureKeyboard

Configure keyboard mappings

ConfigureTerminal

Configure terminal settings

Connect

Establish session

Copy

Copy highlighted text to clipboard

CurrentLocation

Retrieve current cursor position

CustomFont

Select a custom font

DataB

Retrieve binary data

DataCapture

Toggle to start and stop data capture

DeleteConfig

Delete a configuration

Destroy

Destroy emulation window

Disconnect

Close session

GetScreenText

Gets the portion of screen text that is within the specified co-ordinates

GetSelectedText

Gets the portion of text that is highlighted by the user

GetText

Gets the portion of text in the scrollbar buffer that is within the specified co-ordinates

Initialize

Initialize emulation window

ListConfig

List all saved configurations

ModifyConfig

Modify a saved configuration

Paste

Paste text in clipboard

PrinterFont

Select printer fonts

SaveConfig

Save a configuration

SaveTermSetting

Save the terminal settings

SelectPrinter

Select default printer

ScreenCapture

Capture screen data to a file

Transparent

Transparent printer mode

VtFont

Set terminal emulation font

1.6 D_VT220.TXT

The following provides a complete listing of the D_VT220.TXT definition file. If your application uses more than one Distinct ActiveX control in the same form, then some definitions will conflict. For example, the FTP Client ActiveX control includes the definition

```
Global Const ACTION_DISCONNECT = 3
```

in the D_FTP.TXT file and the Telnet ActiveX control includes the definition

```
Global Const ACTION_DISCONNECT = 2
```

in the D_TNET.TXT file. To avoid this conflict, you must rename at least one of the constants (for example, FTP_ACTION_DISCONNECT or TNET_ACTION_DISCONNECT).

```
' VT220 ActiveX Control
' (C) Copyright 1995 - 1998 by Distinct Corporation
' All rights reserved

' actions
Global Const ACTION_NONE = 0
Global Const ACTION_INITIALIZE = 1
Global Const ACTION_CONNECT = 2
Global Const ACTION_DISCONNECT = 3
Global Const ACTION_ABORT = 4
Global Const ACTION_DESTROY = 5

' NumLock Options
Global Const NUMLOCK_NOTSET = 0
Global Const NUMLOCK_LOCKED = 1
Global Const NUMLOCK_NOTLOCKED = 2
```

```
' VT options
Global Const VT_NONE = 0
Global Const VT_LIST_CONFIGURATION = 1
Global Const VT_MODIFY_CONFIGURATION = 2
Global Const VT_DELETE_CONFIGURATION = 3
Global Const VT_SAVE_CONFIGURATION = 4
Global Const VT_CONFIGURE_KEYBOARD = 5
Global Const VT_CONFIGURE_TERMINAL = 6
Global Const VT_CONFIGURE_COLOR = 7
Global Const VT_SET_CUSTOM_FONT = 8
Global Const VT_SET_VTFONT = 9
Global Const VT_SAVE_SETTINGS = 10
Global Const VT_COPY_CLIPBOARD = 11
Global Const VT_PASTE_CLIPBOARD = 12
Global Const VT_CLEAR_CLIPBOARD = 13
Global Const VT_CAPTURE_SCREEN = 14
Global Const VT_CAPTURE_DATA = 15
Global Const VT_PRINTER_SELECT = 16
Global Const VT_PRINTER_FONT = 17
Global Const VT_TRANSPARENT = 18
Global Const VT_CAPTURE_SCREEN_FILE = 19
Global Const VT_CAPTURE_DATA_FILE = 20
Global Const VT_CURRENT_LOCATION = 21

' fonts
Global Const FONT_NONE = 0
Global Const FONT_DEC = 1
Global Const FONT_DEC_SMALL = 2
Global Const FONT_IBM = 3
Global Const FONT_ANSI = 4
Global Const FONT_TERMINAL = 5
Global Const FONT_CUSTOM = 6
Global Const FONT_DEC_MEDIUM = 7
```

```
' error codes
Global Const ERR_CANNOT_GET_SOCKET = 1
Global Const ERR_CANNOT_BIND_SOCKET = 2
Global Const ERR_CANNOT_CHANGE_PORT = 3
Global Const ERR_HOST_UNDEFINED = 4
Global Const ERR_CANNOT_CONNECT = 5
Global Const ERR_IN_ACTION = 6
Global Const ERR_CANNOT_INITIALIZE = 7
Global Const ERR_INITIALIZE_TO_CONNECT = 8
Global Const ERR_INITIALIZE_TO_WRITE = 9
Global Const ERR_CANNOT_DELETE_CONFIGURATION = 10
Global Const ERR_CANNOT_SAVE_CONFIGURATION = 11
Global Const ERR_CANNOT_CAPTURE_SCREEN = 12
Global Const ERR_CANNOT_CAPTURE_DATA = 13
Global Const ERR_CANNOT_CONFIGURE_KEYBOARD = 14
Global Const ERR_CANNOT_CONFIGURE_TERMINAL = 15
Global Const ERR_CANNOT_CONFIGURE_COLOR = 16
Global Const ERR_CANNOT_SET_CUSTOM_FONT = 17
Global Const ERR_CANNOT_SET_VTFONT = 18
Global Const ERR_CANNOT_SAVE_SETTINGS = 19
Global Const ERR_CANNOT_COPY_CLIPBOARD = 20
Global Const ERR_CANNOT_PASTE_CLIPBOARD = 21
Global Const ERR_CANNOT_CLEAR_CLIPBOARD = 22
Global Const ERR_CANNOT_SELECT_PRINTER = 23
Global Const ERR_CANNOT_SET_PRINTER_FONT = 24
Global Const ERR_CANNOT_SET_TRANSPARENT_MODE = 25
Global Const ERR_CANNOT_CAPTURE_SCREEN_FILE = 26
Global Const ERR_CANNOT_CAPTURE_DATA_FILE = 27
Global Const ERR_CANNOT_GET_ROW_COLUMN = 28
Global Const ERR_INVALID_INDEX = 29
Global Const ERR_NO_SELECTED_TEXT = 30
```

2. VT220 Properties

2.1 Action

Summary

Initialize or destroy a VT220 emulation window or connect, disconnect or abort a VT220 session.

Description

The Action property controls the emulation window and the connection state of the VT220 ActiveX control. A session can be established, closed or aborted and the emulation window can be initialized or destroyed by assigning one of the following values to the property.

Value	Meaning
ACTION_INITIALIZE	Initialize emulation window.
ACTION_CONNECT	Establish session.
ACTION_DISCONNECT	Close session.
ACTION_ABORT	Abort session.
ACTION_DESTROY	Destroy emulation window.

This property can be changed at run time only.

The VT220 emulation must be initialized before any other action is performed (with the exception of listing, modifying, deleting and saving configurations as described in the reference page of the VtAction property). To initialize the VT220 emulation, set the Action property to ACTION_INITIALIZE. This will create an emulation window inside the VT220 ActiveX control.

Before setting the Action property to ACTION_CONNECT, the following properties must be initialized. The Host property must be set to the name or internet address (in dotted decimal notation) of the remote server. The Port property must be set to the remote port on which the remote service is running (most servers use the default remote service port 23 for Telnet).

If the connection can be established, then the OnConnect event will be fired. If the connection cannot be established, then the OnError event will be fired. These events will occur before the next statement (i.e. the statement following the assignment of ACTION_CONNECT to the Action property) is executed. The application should set a flag in the OnConnect and OnError events, so that it can determine if the session has been established or not. In addition, the application may want to display an error message in the OnError event to inform the user that the connection could not be established. Please check the reference page of the OnError event for a complete listing of error codes.

Once a connection is no longer needed, the session can be terminated by setting the Action property to ACTION_DISCONNECT. An application must close all connections it has created before it quits. A connection can also be closed by setting the Action property to ACTION_ABORT. This action resets and closes the connection without properly closing down communication and should not be called under normal circumstances.

Before quitting, the application must destroy the emulation window it created with ACTION_INITIALIZE by setting the Action property to ACTION_DESTROY. This will destroy the emulation window and the application can no longer perform any VT220 ActiveX control operations except for ACTION_INITIALIZE and listing, modifying, deleting and saving configurations (see VtAction).

There is no default value for this property.

Note

The Initialize, Connect, Disconnect, Abort and Destroy methods accomplish the same as the above actions. Please check the reference pages of these methods for more detailed information on their usage.

Example

```
Vt220Client.Action = ACTION_INITIALIZE
Vt220Client.Host = "speedy.distinct.com"
Vt220Client.Port = 23
Vt220Client.Action = ACTION_CONNECT
```

2.2 BinaryData**Summary**

Contains the binary data received following a call to the DataB method.

Description

After each call to the DataB method in a C# application this property needs to be read to access binary data.

Example

```
private void VT220_OnDataB(object sender, AxD_VtLib._DVtEvents_OnDataBEvent e)
{
    int bytes;
    int len;
    bool result;

    bytes = e.bytes;
    Object arrData = new byte[bytes];
    len = axVT2201.DataB (arrData, bytes);
    if (len > 0)
    {
        arrData = axVT2201.BinaryData;
        result = axSocket1.SendB (arrData, len);
    }
}
```

2.3 CaptureAppend

Summary

Specifies whether captured data should be appended to or overwritten in the file specified by the CaptureFile property.

Description

The boolean CaptureAppend property determines whether to overwrite or append incoming data to CaptureFile. An application may want to overwrite or append data to CaptureFile depending on user input.

If this property is set to True, then the data captured will be appended to the file (specified by the CaptureFile property). Otherwise, the contents of the file (specified by the CaptureFile property) will be overwritten.

This property is only used when the VtAction property is set to VT_CAPTURE_DATA_FILE or VT_CAPTURE_SCREEN_FILE.

The default value for this property is False. This property can be read or written to at any time.

Example

```
Vt220Client.CaptureAppend = True
```

2.4 CaptureData

Summary

This property is used to check if it is currently possible to capture incoming data to a file.

Description

The boolean CaptureData property is used to check if it is possible to start capturing incoming data to a disk file at that specific time. An application may want to enable or disable a menu item depending on the value of the CaptureData property. The InCaptureData property can be used to check if the capture of incoming data is currently enabled or disabled.

To start data capture, the DataCapture method must be called with the *CaptureFile* parameter set to an empty string (or the VtAction property must be set to VT_CAPTURE_DATA). To stop the data capture, the DataCapture method must be called again with the *CaptureFile* parameter set to an empty string (or the VtAction property must again be set to VT_CAPTURE_DATA). Any data that was displayed on the VT220 emulation window between these two actions will be captured and saved to a disk file.

There is no default value for this property. This property can only be read at run time after the emulation has been initialized and cannot be written to at any time.

Example

```
Menu_Edit_Item(5).Enabled = (Vt220Client.CaptureData = True)
```

2.5 CaptureDataFile

Summary

This property is used to check if it is currently possible to capture incoming data to the file specified in the CaptureFile property.

Description

The boolean CaptureDataFile property is used to check if it is possible to start capturing incoming data to the file specified by the CaptureFile property at that specific time. An application may want to enable or disable a menu item depending on the value of the CaptureDataFile property. The InCaptureDataFile property can be used to check if the capture of incoming data is currently enabled or disabled.

To start data capture to a specific file, the DataCapture method must be called with the *CaptureFile* parameter set to a valid file name (or the VtAction property must be set to VT_CAPTURE_DATA_FILE). To stop the data capture to a specific file, the DataCapture method must be called again with the *CaptureFile* parameter set to the file name used in the first call (or the VtAction property must again be set to VT_CAPTURE_DATA_FILE). Any data that was displayed on the VT220 emulation window between these two actions will be captured and saved to the file specified by the CaptureFile property. The CaptureAppend property determines if the contents of this file will be overwritten or appended with the captured data.

There is no default value for this property. This property can only be read at run time after the emulation has been initialized and cannot be written to at any time.

Example

```
Menu_Edit_Item(7).Enabled = (Vt220Client.CaptureDataFile = True)
```

2.6 CaptureFile

Summary

Specifies the name of the file in which the captured data will be stored.

Description

The CaptureFile property specifies the file name for the captured data to be stored. This property must be set before calling the CaptureData method (or setting the VtAction property to VT_CAPTURE_DATA_FILE) or before calling the ScreenCapture method (or setting the VtAction property to VT_CAPTURE_SCREEN_FILE.) The CaptureAppend property determines if the contents of this file will be overwritten or appended with the captured data.

Check the CaptureDataFile and CaptureScreenFile properties before actually starting a data or screen capture. Also, check the InCaptureDataFile property to see if a data capture is in progress.

The default value for this property is an empty file name. This property can be read or written to at any time.

Example

```
Vt220Client.CaptureFile = "screen.txt"  
Vt220Client.VtAction = VT_CAPTURE_SCREEN_FILE
```

2.7 CaptureScreen

Summary

This property is used to check if it is currently possible to capture the screen contents to a file.

Description

The boolean CaptureScreen property is used to check if it is possible to capture the contents of the screen to a disk file at that specific time. An application may want to enable or disable a menu item depending on the value of the CaptureScreen property.

To capture the current screen, the ScreenCapture method must be called with the *CaptureFile* parameter set to a valid file name (or the VtAction property must be set to VT_CAPTURE_SCREEN). This will capture the current screen contents of the VT220 emulation window to a disk file.

There is no default value for this property. This property can only be read at run time after the emulation has been initialized and cannot be written to at any time.

Example

```
Menu_Edit_Item(4).Enabled = (Vt220Client.CaptureScreen = True)
```

2.8 CaptureScreenFile

Summary

This property is used to check if it is currently possible to capture the screen data to the file specified by the CaptureFile property.

Description

The boolean CaptureScreenFile property is used to check if it is possible to capture the contents of the screen to the file specified by the CaptureFile property at that specific time. An application may want to enable or disable a menu item depending on the value of the CaptureScreenFile property.

To capture the current screen, call the ScreenCapture method with the *CaptureFile* parameter set to a valid file name(or set the VtAction property to VT_CAPTURE_SCREEN_FILE.) This will capture the current screen contents of the VT220 emulation window to the file specified by the CaptureFile property. The CaptureAppend property determines if the contents of this file will be overwritten or appended with the captured data.

There is no default value for this property. This property can only be read at run time after the emulation has been initialized and cannot be written to at any time.

Example

```
Menu_Edit_Item(6).Enabled = (Vt220Client.CaptureScreenFile = True)
```

2.9 ClearClipboard

Summary

This property is used to check if it is currently possible to clear the clipboard text.

Description

The boolean ClearClipboard property is used to check if it is possible to clear the clipboard text at that specific time. If the clipboard does not contain any text at this point, then the clipboard cannot be cleared. An application may want to enable or disable a menu item depending on the value of the ClearClipboard property.

The clipboard is used to temporarily store text (among other data formats) that is copied to it from Windows applications. The clipboard will retain the text until another text selection is copied to it. To clear the text in the clipboard, call the Clear method (or set the VtAction property to VT_CLEAR_CLIPBOARD.)

There is no default value for this property. This property can only be read at run time after the emulation has been initialized and cannot be written to at any time.

Example

```
Menu_Edit_Item(3).Enabled = (Vt220Client.ClearClipboard = True)
```

2.10 Column

Column position of the cursor.

Description

The Column property will contain the current column position of the cursor after the CurrentLocation method is called (or after the VtAction property is set to VT_CURRENT_LOCATION). If the action is successful, the Column property will contain the current column position of the cursor; otherwise, the OnError event will occur before the next line of code is reached.

The Column property should be read only after calling the CurrentLocation method (or setting the VtAction property to VT_CURRENT_LOCATION). The application should check that no error has occurred to ensure that the cursor position information is accurate.

There is no default value for this property. This property can be read at any time.

Example

```
Vt220Client.VtAction = VT_CURRENT_LOCATION
If NoError Then
    Column = Vt220Client.Column
End If
```

2.11 ConfigurationName

Summary

Specifies the name of the VT220 configuration profile to be modified, saved or deleted.

Description

The ConfigurationName property specifies the name of the VT220 terminal emulation settings currently in use. Any action affecting the terminal settings will modify the values of the configuration profile specified by this property.

The ConfigurationName property must be set to the name of the VT220 terminal configuration profile to be saved before calling the SaveConfig method (or setting VtAction to VT_SAVE_SETTINGS.) The property must be set to specify a configuration name before the application initializes the VT220 emulation by calling the Initialize method (or by setting the Action property to ACTION_INITIALIZE.) Before the application can modify terminal emulation settings by setting the VtAction property to VT_MODIFY_CONFIGURATION or before deleting or saving a configuration by setting the VtAction property to VT_DELETE_CONFIGURATION or VT_SAVE_CONFIGURATION, the ConfigurationName property must also contain the name of a configuration.

There is no default value for this property. This property can be set or read at any time.

Example

```
Vt220Client.ConfigurationName = "test"  
Vt220Client.VtAction = VT_MODIFY_CONFIGURATION
```

2.12 CopyClipboard

Summary

Specifies whether it is possible to copy text to the clipboard.

Description

The boolean CopyClipboard property is used to check if it is possible to copy text to the clipboard at that specific time. The CopyClipboard property will be False if the user has not selected any text in the VT220 emulation window. An application may want to enable or disable a menu item depending on the value of the CopyClipboard property.

The clipboard is used to temporarily store text (among other data formats) that is copied to it from Windows applications. The clipboard will retain the text until another text selection is copied to it. An application can clear the text contained in the clipboard using the Clear method (or setting VT_CLEAR_CLIPBOARD action) or copy text to the clipboard using the Copy method (or setting the VT_COPY_CLIPBOARD action).

There is no default value for this property. This property can only be read at run time after the emulation has been initialized and cannot be written to at any time.

Example

```
Menu_Edit_Item(1).Enabled = (Vt220Client.CopyClipboard = True)
```

2.13 CurrentFont

Summary

Name of font that is currently in use.

Description

The CurrentFont property specifies the font currently in use by the VT220 emulation window. The property is also used to specify the new font to be used before setting the VtAction property to VT_SET_VTFONT. The CurrentFont property can be set to one of the following values.

Value	Meaning
FONT_DEC	DEC large font.
FONT_DEC_SMALL	DEC small font.
FONT_IBM	IBM font.
FONT_ANSI	ANSI font.
FONT_TERMINAL	VT terminal font.
FONT_CUSTOM	Custom font.
FONT_DEC_MEDIUM	DEC medium font.

There is no default value for this property. This property can be read at any time to determine the current font in use.

Example

```
Vt220Client.CurrentFont = FONT_DEC  
Vt220Client.VtAction = VT_SET_VTFONT
```

2.14 EnableClipboard

Summary

Enables or disables the use of the mouse buttons for accessing the Windows clipboard.

Description

The EnableClipboard property allows you to enable or disable the mouse's ability to transfer text to and from the Windows clipboard.

The default setting for this property is True. This means the left mouse can be used to select and copy text to the clipboard and the right mouse button can be used to paste that text to a location on the VT220 screen. To disable the transfer of text to and from the clipboard, set EnableClipboard to False. This property can be set at any time.

Example

```
Vt220Client.EnableClipboard = False
```

2.15 Host

Summary

Specifies the name or dotted decimal internet address of the remote host to which a connection will be made .

Description

The Host property specifies the name or internet address of a remote server. This property must be set before a session can be established. There are three possible ways to specify a remote server.

Machine Name

An application only needs to specify the name of the remote server if the server is located on the same network as the local PC or if the internet address of the server is defined in the local host table. If the remote server is not on the local network, then the underlying protocol will route the traffic through a gateway. If the remote server is not defined in the local host table, then the underlying protocol will contact the domain server to resolve the internet address of the server.

Machine and Domain Name

An application needs to specify the machine name and the domain name if the remote server is not located on the same network as the local PC. Fully domain qualified machine names are written from left to right in ascending order (for example, *speedy.distinct.com*). If both machine and domain names are specified, then the underlying protocol will contact the domain server to resolve the internet address of the server.

Internet Address

Sometimes the user knows only the internet address of the remote server that he or she wants to use. In this case, the internet address can be entered in what is known as the dotted decimal notation (for example, *127.43.101.12*). If the remote server identified by this address is not on the local network, then the underlying protocol will route the traffic through a gateway.

This property can be changed at design time and at run time before a connection has been established. There is no default value for this property.

Example

```
Vt220Client.Action = ACTION_INITIALIZE
Vt220Client.Host = "127.43.101.12"
Vt220Client.Port = 23
Vt220Client.Action = ACTION_CONNECT
```

2.16 InCaptureData

Summary

Check if data capture is enabled.

Description

The boolean `InCaptureData` property is used to check if incoming data is currently being captured to a disk file. An application may want to check or uncheck a menu item depending on the value of the `InCaptureData` property. The `CaptureData` property can be used to check if it is possible to start capturing incoming data to a disk file at that specific time.

To start data capture, the `DataCapture` method must be called with the *CaptureFile* parameter set to an empty string (or the `VtAction` property must be set to `VT_CAPTURE_DATA`). To stop the data capture, the `DataCapture` method must be called again with the *CaptureFile* parameter set to an empty string (or the `VtAction` property must again be set to `VT_CAPTURE_DATA`). Any data that was displayed on the VT220 emulation window between these two actions will be captured and saved to a disk file.

There is no default value for this property. This property can only be read at run time after the emulation has been initialized and cannot be written to at any time.

Example

```
If Vt220Client.InCaptureData = True Then
    MsgBox "Data capture is in progress", 64, "Sample Program"
End If
```

2.17 InCaptureDataFile

Summary

Check if data capture to CaptureFile is enabled.

Description

The boolean InCaptureDataFile property is used to check if incoming data is currently being captured to CaptureFile. An application may want to check or uncheck a menu item depending on the value of the InCaptureDataFile property. The CaptureDataFile property can be used to check if it is possible to start capturing incoming data to CaptureFile at that specific time.

To start data capture, the DataCapture method must be called with the *CaptureFile* parameter set to a valid file name (or the VtAction property must be set to VT_CAPTURE_DATA_FILE). To stop the data capture, the DataCapture method must be called again with the *CaptureFile* parameter set to the file name used in the first call (or the VtAction property must again be set to VT_CAPTURE_DATA_FILE). Any data that was displayed on the VT220 emulation window between these two actions will be captured and saved to the file specified by the CaptureFile property.

There is no default value for this property. This property can only be read at run time after the emulation has been initialized and cannot be written to at any time.

Example

```
If Vt220Client.InCaptureDataFile = True Then
    MsgBox "Data capture to CaptureFile is in progress", 64, "Sample Program"
End If
```

2.18 NumLockMode (Windows 95 only)

Summary

Set the behavior of the NumLock key on the keyboard.

Description

The NumLockMode property controls the behavior of the NumLock key on the numeric keypad. The behavior can be set by assigning one of the following values to the property.

Value	Meaning
NUMLOCK_NOTSET	The end-user can choose the status of the NumLock key.
NUMLOCK_LOCKED	VT220 control forces the NumLock key to stay on.
NUMLOCK_UNLOCKED	VT220 control forces the NumLock key to stay off.

This property can be changed at run time or design time.

The default setting for this property is NUMLOCK_NOTSET.

Example

```
Vt220Client.NumLockMode = NUMLOCK_LOCKED
```

2.19 PasteClipboard

Summary

Specifies if it is currently possible to paste text from clipboard to the VT emulation window.

Description

The boolean PasteClipboard property is used to check if it is possible to paste the text selection contained in the clipboard into the VT220 emulation window at that specific time. The PasteClipboard property will be False if the user has not copied any text into the clipboard. An application may want to enable or disable a menu item depending on the value of the PasteClipboard property.

The clipboard is used to temporarily store text (among other data formats) that is copied to it from Windows applications. The clipboard will retain the text until another text selection is copied to it. An application can clear the text contained in the clipboard using the Clear method (or the VT_CLEAR_CLIPBOARD action) or paste text from the clipboard using the Paste method (or the VT_PASTE_CLIPBOARD action). Please refer to the VtAction property for more details.

There is no default value for this property. This property can only be read at run time while a connection is established.

Example

```
Menu_Edit_Item(2).Enabled = (Vt220Client.PasteClipboard = True)
```

2.20 Port

Summary

Specifies the telnet service port on the remote host to which a connection will be established.

Description

The Port property specifies the remote port on the remote server on which the Telnet service resides. Most Telnet services listen for connection requests on port 23. Sometimes, for security reasons, port numbers other than 23 are used for remote connections.

If the application will be connecting to remote services on a different port, then the Port property must be set before the connection attempt is made. An application may even want to query the user for the correct port before connecting. The ActiveX control does not verify the setting of the Port property and any value is therefore legal.

This property can be changed at design time and at run time before a connection is established. The default value for this property is port 23.

Example

```
Vt220Client.Action = ACTION_INITIALIZE  
Vt220Client.Host = "speedy.distinct.com"  
Vt220Client.Port = 23  
Vt220Client.Action = ACTION_CONNECT
```

2.21 Row

Summary

Row position of the cursor.

Description

The Row property will contain the current row position of the cursor after the CurrentLocation method is called (or the VtAction property is set to VT_CURRENT_LOCATION). If the action is successful, the Row property will contain the current row position of the cursor; otherwise, the OnError event will occur before the next line of code is reached.

The Row property should be read only after calling the CurrentLocation method (or setting the VtAction property to VT_CURRENT_LOCATION). The application should check that no error has occurred to ensure that the cursor position information is accurate.

There is no default value for this property. This property can be read at any time.

Example

```
Vt220Client.VtAction = VT_CURRENT_LOCATION
If NoError Then
    Row = Vt220Client.Row
End If
```

2.22 SaveSettings

Summary

Specifies if it is currently possible to save the configuration settings.

Description

The boolean SaveSettings property is used to check if it is possible to save the terminal settings at that specific time. An application may want to enable or disable a menu item depending on the value of the SaveSettings property.

To save the terminal settings, call the SaveTermSetting method (or set the VtAction property to VT_SAVE_SETTINGS). The terminal settings will be saved and associated with the configuration name specified by the user. Before terminal settings can be saved the Initialize method must be called (or the Action property set to ACTION_INITIALIZE).

There is no default value for this property. This property can only be read at run time after the emulation has been initialized and cannot be written to at any time.

Example

```
Menu_Action_Item(4).Enabled = (Vt220Client.SaveSettings = True)
```

2.23 ShowHelp

Summary

Display or hide dialog box help buttons.

Description

The ShowHelp property is used to enable or disable the display of help buttons in the dialog boxes displayed by the VT220 ActiveX control such as the keyboard dialog box. By default, all VT220 ActiveX control dialog boxes used to obtain input from the user display a help button. Once the user clicks on any one of the help buttons, the ActiveX control displays a section of its help file. If an application wants to disable the help buttons to avoid the display of the ActiveX control's help file, then it can set the ShowHelp property to False.

The ShowHelp property can be changed at any time. This allows an application to selectively enable and disable help buttons just before a specific dialog box is displayed.

This property can be set at any time. The default value for this property is True.

This property allows you to create your own customized help for each ActiveX built-in dialog. If you do not wish to have a help file you should set this property to FALSE and the help button will not appear on these dialog boxes. On the other hand if you wish to create a help file you need to name it **VT220.hlp** and it should be in the same directory as the VT220 ActiveX control.

Listed below are the dialogs that have a help button. For each of the respective dialogs help topics use the associated topic id:

Dialog Box	Help Topic Id to use
Terminal Configuration	100
Terminal Edit	101
Save Configuration	102
Terminal Color	103
Terminal Font	104
Keyboard Mapping	105
Save Key Mapping File	106
Print Setup	107
Print Font	108
Screen Capture	109
Data Capture	110
Load Key Map File	111

Example

Vt220Client.**ShowHelp** = False

2.24 ShowScrollbars

Summary

Displays or disables the scrollbars in the VT emulation window.

Description

The ShowScrollbars property is used to enable or disable the scrollbars in the presentation space created by the VT220 ActiveX control. By default, the VT220 ActiveX control will display horizontal and vertical scrollbars. If an application wants to disable the display of scrollbars, then it can set the ShowScrollbars property to False.

The ShowScrollbars property can be changed at any time. This allows an application to selectively enable and disable the scrollbars.

This property can be set at any time. The default value for this property is True.

Example

```
Vt220Client.ShowScrollbars = False
```

2.25 TransModeStatus

Summary

Transparent mode status.

Description

The boolean TransModeStatus property is used to check the current status of the printer transparent mode. An application may want to check or uncheck a menu item depending on the value of the TransModeStatus property.

To turn on the printer transparent mode, set the VtAction property to VT_TRANSPARENT (or call the Transparent method). Then, the TransModeStatus property will be True, indicating that the transparent mode is on. To turn off the transparent mode, set the VtAction property to VT_TRANSPARENT (or call the Transparent method) once again. Then, the TransModeStatus property will be FALSE, indicating that the transparent mode is off.

There is no default value for this property. This property can only be read at run time after the emulation has been initialized and cannot be written to at any time.

Example

```
Menu_Printer_Item(2).Checked = (Vt220Client.TransModeStatus = True)
```

2.26 UseVariant

Summary

Specifies whether data is to be received in binary or ascii format.

Description

The UseVariant property is used to specify whether data is to be received in binary or ascii form. If this property is set to True, the OnDataB event will be fired when data arrives; otherwise, the OnData event is fired.

This property should be set before any data arrives or before sending any data. This property can be read at any time. The default value for this property is False.

Example

```
VT220Client.UseVariant = True
```

2.27 WriteData

Summary

Display data in VT220 emulation window.

Description

The WriteData property is used to display data in the VT220 terminal emulation window. Data is displayed by assigning a string to this property. Normally, an application will display incoming data from the server in the emulation window by assigning the string containing the incoming data to this property. The ActiveX control takes care of all emulation required to correctly display the data.

If an application handles the connection itself, then it must pass incoming data to the ActiveX control in this manner. If the ActiveX control is used to handle the connection, then an application may still add messages to the VT220 emulation window by assigning them to this property. This allows the application to display progress information, such as "Trying..." or "Connected", in the VT220 emulation window.

This property can be set at run time after the VT220 emulation window has been initialized by calling the Initialize method (or setting the Action property to ACTION_INITIALIZE.) There is no default value for this property.

Note

The WriteData property can be used to insert any sequence of data including control sequences to change an option. For example to set the reverse video option you could set this property as follows:

```
Vt220Client.WriteData = Chr(27) + "[m" + Chr(27) + "[7m"
```

Example

```
Sub VT220Client_OnConnect ()  
    Vt220Client.WriteData = "Connected."  
    Connected = True  
End Sub
```

2.28 VtAction

Summary

Set up terminal, color, font, keyboard or printer and list, save, modify or delete configurations.

Description

The VtAction property controls the emulation features of the VT220 ActiveX control. The following values can be assigned to the VtAction property.

Value	Meaning
VT_LIST_CONFIGURATION	List all saved configurations.
VT_MODIFY_CONFIGURATION	Modify a saved configuration.
VT_DELETE_CONFIGURATION	Delete a configuration.
VT_SAVE_CONFIGURATION	Save a configuration.
VT_CONFIGURE_KEYBOARD	Configure keyboard mappings.
VT_CONFIGURE_TERMINAL	Configure terminal settings.
VT_CONFIGURE_COLOR	Set background and foreground colors.
VT_SET_CUSTOM_FONT	Select a custom font.
VT_SET_VTFONT	Set the terminal emulation font.
VT_SAVE_SETTINGS	Save the terminal settings.
VT_COPY_CLIPBOARD	Copy highlighted text into buffer.
VT_PASTE_CLIPBOARD	Paste text in clipboard.
VT_CLEAR_CLIPBOARD	Clear text in clipboard.
VT_CAPTURE_SCREEN	Capture screen data into a file.
VT_CAPTURE_DATA	Toggle to start and stop data capture.
VT_PRINTER_SELECT	Select default printer.
VT_PRINTER_FONT	Select printer fonts.
VT_TRANSPARENT	Transparent printer mode.
VT_CAPTURE_SCREEN_FILE	Capture screen data to CaptureFile.
VT_CAPTURE_DATA_FILE	Toggle to start and stop data capture to CaptureFile.

The current VT220 terminal settings (for example, inverse video) can be changed by setting the VtAction property to VT_CONFIGURE_TERMINAL. To map the keys on the keyboard, set the VtAction property to VT_CONFIGURE_KEYBOARD. The foreground, background and bold colors of the VT220 terminal can be changed by setting the VtAction property to VT_CONFIGURE_COLOR. To select a custom font, set the VtAction property to VT_SET_CUSTOM_FONT. These actions will cause the ActiveX control to bring up dialog boxes, which allow the user to select new settings.

To select the type of font to be used for VT220 emulation, set the CurrentFont property to the appropriate value and then set the VtAction property to VT_SET_VTFONT. To save the current state of the VT220 terminal emulation window, such as font, color, terminal settings and keyboard mappings, set the VtAction property to VT_SAVE_SETTINGS.

Some of the actions can be invoked before the VT220 terminal emulation window is initialized. The application can change the terminal configuration to be used before the initialization of the emulation window by setting the VtAction property to VT_MODIFY_CONFIGURATION. The ConfigurationName property must previously be set to a valid configuration name. A list of the names of all available configurations can be obtained by setting the VtAction property to VT_LIST_CONFIGURATION. The list of configuration names will be delivered to the application in one or more OnList events. To delete a configuration, set the ConfigurationName property to a valid configuration name and then set the VtAction property to VT_DELETE_CONFIGURATION.

To save a configuration, set the `ConfigurationName` property to an existing or a new configuration name and then set the `VtAction` property to `VT_SAVE_CONFIGURATION`.

Clipboard activities, such as copy, paste and clear, can be performed with the `VtAction` property. Copying the selected text in the VT220 emulation window to the clipboard can be accomplished by setting the `VtAction` property to `VT_COPY_CLIPBOARD`. Before doing this, an application must make sure that this action can be performed by verifying that the `CopyClipboard` property is `True`. Pasting text contained in the clipboard to the VT220 emulation window is done by setting the `VtAction` property to `VT_PASTE_CLIPBOARD`. Before doing this, an application must make sure that this action can be performed by verifying that the `PasteClipboard` property is set to `True`. An application can clear the text contained in the clipboard by setting the `VtAction` property to `VT_CLEAR_CLIPBOARD`. This action is only legal if the `ClearClipboard` property is `True`.

To capture the data currently displayed on the screen of the VT220 emulation window to a disk file, set the `VtAction` property to `VT_CAPTURE_SCREEN`. Before doing this, an application must make sure that this action can be performed by verifying that the `CaptureScreen` property is `True`. To capture all incoming data to a disk file, set the `VtAction` property to `VT_CAPTURE_DATA`. This action is only legal if the `CaptureData` property is `True`. Capturing incoming data to a disk file is disabled by setting the `VtAction` property to `VT_CAPTURE_DATA` a second time. An application can check the value of the boolean `InCaptureData` property to determine if the VT220 ActiveX control is currently capturing incoming data to a disk file. Both the `VT_CAPTURE_SCREEN` and the `VT_CAPTURE_DATA` action will display a dialog box to obtain the name of the destination disk file from the user.

To capture the data currently displayed on the screen of the VT220 emulation window to a file specified by the `CaptureFile` property, set the `VtAction` property to `VT_CAPTURE_SCREEN_FILE`. Before doing this, an application must make sure that this action can be performed by verifying that the `CaptureScreenFile` property is `True`. To capture all incoming data to `CaptureFile`, set the `VtAction` property to `VT_CAPTURE_DATA_FILE`. This action is only legal if the `CaptureDataFile` property is `True`. Capturing incoming data to `CaptureFile` is disabled by setting the `VtAction` property to `VT_CAPTURE_DATA_FILE` a second time. An application can check the value of the boolean `InCaptureDataFile` property to determine if the VT220 ActiveX control is currently capturing incoming data to `CaptureFile`.

The `VtAction` property can also be used to configure the printer to be used with the VT220 ActiveX control. Setting the `VtAction` property to `VT_PRINTER_SELECT` will select the default printer to be used. The printer font can be selected by setting the `VtAction` property to `VT_PRINTER_FONT` and the selected printer font can be replaced with the default printer font by setting the `VtAction` property to `VT_TRANSPARENT`. This essentially sends all keyboard inputs to the printer. Setting `VT_TRANSPARENT` to the `VtAction` property a second time turns off the printing.

The `ListConfig`, `ModifyConfig`, `DeleteConfig`, `SaveConfig`, `ConfigureKeyboard`, `ConfigureTerminal`, `ConfigureColor`, `CustomFont`, `VtFont`, `SaveTermSetting`, `Copy`, `Paste`, `Clear`, `ScreenCapture`, `DataCapture`, `SelectPrinter`, `PrinterFont` and `Transparent` methods accomplish the same as the above actions. Please check the reference pages of these methods for more detailed information on their usage.

This property can only be set at run time after initializing the VT220 emulation window by setting the `Action` property to `ACTION_INITIALIZE`. There is no default value for this property.

Example

```
Vt220Client.VtAction = VT_COPY_CLIPBOARD
```


3_VT220 Events

3.1 OnClose

Summary

Remote connection has been closed.

Description

The OnClose event occurs usually in response to calling the Disconnect or Abort methods (or setting the Action property to ACTION_DISCONNECT or ACTION_ABORT). In some cases, the remote server may close a connection (for example because of a long period of inactivity). This will also trigger an OnClose event. In this case, the application must still call the Disconnect method (or set the Action property to ACTION_DISCONNECT) to free up all the resources allocated for the connection. However, this should not be done during the OnClose event because it might result in an infinite loop.

If this event occurs in response to calling the Disconnect method (or setting the Action property to ACTION_DISCONNECT), it will occur before next line of code is executed.

Normally, an application should simply set a flag in response to this event. Then, this flag can be checked directly after the Disconnect method (or the ACTION_DISCONNECT action) to make sure that the connection was actually terminated.

Example

```
Sub VT220Client_OnClose ()
  If Connected Then
    ' disconnect and destroy VT220 terminal
    Connected = False
    Vt220Client.WriteData = "Connection Closed."
    Vt220Client.Action = ACTION_DISCONNECT
    Vt220Client.Action = ACTION_DESTROY
  End If
End Sub
```

3.2 OnConnect

Summary

Remote connection has been established.

Description

The OnConnect event occurs in response to the Connect method (or in response to setting the Action property to ACTION_CONNECT). This event will occur before the next line of code is executed.

Normally, an application should simply set a flag in response to this event. Then, this flag can be checked directly after the Connect method (or the ACTION_CONNECT action) to make sure that the connection was actually established.

If the connection could not be established, then the OnError event will be called instead of the OnConnect event.

While handling the OnConnect event, an application should not perform tasks that have the potential of requiring a lot of time to complete, such as generating a message box.

Example

```
Sub VT220Client_OnConnect ()  
    Vt220Client.WriteData = "Connected."  
    Connected = True  
End Sub
```

3.3 OnData

Summary

Data from VT220 emulation window to be sent to remote server.

Description

The OnData event occurs when the VT220 ActiveX control needs to deliver data to the remote server. This event is usually caused by user input (key strokes). This event will only occur if the application is handling the connection itself.

Normally, the application should transmit the data delivered by the ActiveX control during this event directly to the remote server. Sometimes it may be necessary for the application to filter or change user input before it is delivered to the server (for example, to expand macros).

Data received by the application from the remote server should be sent directly to the VT220 ActiveX control by assigning the data to the WriteData property. Sometimes an application may need to scan the incoming data, for example, to look for a password prompt during login.

While handling the OnData event, an application should not perform tasks that have the potential of requiring a lot of time to complete, such as generating a message box.

Example

```
Sub VT220Client_OnData (Buffer As String)
    Socket.Send = Buffer           ' where Socket is the Windows Sockets ActiveX control
End Sub
```

3.4 OnDataB

Summary

Data from VT220 emulation window to be sent to remote server as binary.

Description

The OnDataB event occurs when the VT220 ActiveX control needs to deliver data to the remote server. This event is usually caused by user input (key strokes). This event will only occur if the application is handling the connection itself.

The event notifies the application how many bytes of data is ready to be retrieved. It is up to the application to call the DataB method to retrieve and process the data. Normally, the application should transmit the data retrieved using the DataB method during this event directly to the remote server. Sometimes it may be necessary for the application to filter or change user input before it is delivered to the server (for example, to expand macros).

Data received by the application from the remote server should be sent directly to the VT220 ActiveX control by assigning the data to the WriteData property. Sometimes an application may need to scan the incoming data, for example, to look for a password prompt during login.

While handling the OnDataB event, an application should not perform tasks that have the potential of requiring a lot of time to complete, such as generating a message box.

Example

```
Sub VT220Client_OnDataB (Bytes As Long)
  Dim Buffer(1 To Bytes) As Byte
  Dim Siz As Long

  Siz = VT220Client.DataB (Buffer, Bytes)
  If Siz > 0 Then
    Result = Socket.SendB (Buffer, Siz) ' where Socket is Windows Sockets ActiveX control
  End If
End Sub
```

3.5 OnError

Summary

Local error has occurred.

Description

The OnError event occurs when a property is accessed in an illegal way or when a connection with the remote server cannot be established. The table below lists all possible error codes delivered by this event.

Value	Meaning
ERR_CANNOT_GET_SOCKET	Unable to allocate socket.
ERR_CANNOT_BIND_SOCKET	Unable to bind socket to local address.
ERR_CANNOT_CHANGE_PORT	Cannot change port while connected. An attempt was made to change the Port property while connected. Disconnect by calling the Disconnect method (or setting the Action property to ACTION_DISCONNECT) before trying to change the Port property.
ERR_HOST_UNDEFINED	Remote host must be defined before connecting. The Host property must be defined before calling the Connect method (or setting the Action property to ACTION_CONNECT).
ERR_CANNOT_CONNECT	Unable to connect to remote host. The Port or the Host properties may be set incorrectly or the host may be down.
ERR_IN_ACTION	Another action is in progress.
ERR_CANNOT_INITIALIZE	Cannot initialize the emulation window.
ERR_INITIALIZE_TO_CONNECT	You must initialize the emulation window, by calling the Initialize method, before connecting.
ERR_INITIALIZE_TO_WRITE	Initialize emulation window before writing data to the emulation window.
ERR_CANNOT_DELETE_CONFIGURATION	Unable to delete the specified configuration. The ConfigurationName property may be set to an invalid configuration name. Set the ConfigurationName property before calling the DeleteConfig method (or setting the VtAction property to VT_DELETE_CONFIGURATION).
ERR_CANNOT_SAVE_CONFIGURATION	Unable to save the specified configuration. The ConfigurationName property may be set to an invalid configuration name. Set the ConfigurationName property before calling the SaveConfig method (or setting the

ERR_CANNOT_CAPTURE_SCREEN	VtAction property to VT_SAVE_CONFIGURATI ON). Unable to capture data on screen to a file. The VT220 ActiveX control could not open or create the file to store the data on the screen.
ERR_CANNOT_CAPTURE_DATA	Unable to capture incoming data to a file. The VT220 ActiveX control could not open or create the file to store the incoming data.
ERR_CANNOT_CONFIGURE_KEYBOARD	Unable to configure keyboard.
ERR_CANNOT_CONFIGURE_TERMINAL	Unable to configure terminal.
ERR_CANNOT_CONFIGURE_COLOR	Unable to configure color.
ERR_CANNOT_SET_CUSTOM_FONT	Unable to set custom terminal font.
ERR_CANNOT_SET_VTFONT	Unable to set VT font.
ERR_CANNOT_SAVE_SETTINGS	Unable to save configuration settings.
ERR_CANNOT_COPY_CLIPBOARD	Unable to copy text to clipboard.
ERR_CANNOT_PASTE_CLIPBOARD	Unable to paste text from clipboard.
ERR_CANNOT_CLEAR_CLIPBOARD	Unable to clear clipboard text.
ERR_CANNOT_SELECT_PRINTER	Unable to select a default printer.
ERR_CANNOT_SET_PRINTER_FONT	Unable to set a default printer font.
ERR_CANNOT_SET_TRANSPARENT_MODE	Unable to set printer transparent mode.
ERR_CANNOT_CAPTURE_SCREEN_FILE	Unable to capture data on screen to CaptureFile. The VT220 ActiveX control could not open or create the file specified by the CaptureFile property to store the data on the screen
ERR_INVALID_INDEX	There is an error in the coordinates specified by the GetText method.
ERR_CANNOT_CAPTURE_DATA_FILE	Unable to capture incoming data to CaptureFile. The VT220 ActiveX control could not open or create the file specified by the CaptureFile property to store the incoming data.

Example

```
Sub VT220Client_OnError (ErrorCode As Integer)
    If ErrorCode = ERR_CANNOT_CONNECT Then
        MsgBox "Unable to connect to remote host", 64, "Sample Program"
    End If
End Sub
```

3.6 OnList

Summary

Listing of configuration names.

Description

The OnList event occurs when the ListConfig method is called (or the VtAction property is set to VT_LIST_CONFIGURATION). This event will only occur if the name of at least one configuration is available. One or more OnList events will occur to deliver the complete list, where each OnList event will deliver exactly one configuration name.

Normally, an application should simply display each configuration name in a list box or combo box. Each name delivered by the OnList event can be added to the list box or combo box. Then, the user can select the name of the configuration settings to be used for the next session.

While handling the OnList event, an application should not perform tasks that have the potential of requiring a lot of time to complete, such as generating a message box.

Example

```
Sub VT220Client_OnList (Entry As String)
    ListBox.AddItem Entry
End Sub
```

4 VT220 Methods

4.1 Abort

Summary

Abort VT220 session.

Syntax

Boolean Abort ()

Description

The Abort method aborts a VT220 session. This method resets and closes the connection without properly closing down communication and should not be called under normal circumstances.

The Abort method takes no parameters and returns a boolean. If the connection is successfully reset and closed, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value.

Calling this method is equivalent to setting the Action property to ACTION_ABORT.

Example

```
Result = VT220Client.Abort ()  
If Result = False Then  
    MsgBox "Unable to abort", 64, "Sample Program"  
End If
```

4.2 Clear

Summary

Clear clipboard contents.

Syntax

Boolean Clear ()

Description

Clipboard activities, such as copy and paste, can be performed in the emulation window. An application can clear the text contained in the clipboard by calling the Clear method. This action is only legal if the ClearClipboard property is True. Any selected text in the emulation window can be copied to the clipboard using the Copy method, and the Paste method can be used to paste the contents of the clipboard into the emulation window.

The Clear method takes no parameters and returns a boolean. If the clipboard can be successfully cleared, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value.

Calling this method is equivalent to setting the VtAction property to VT_CLEAR_CLIPBOARD. This method can only be called after the VT220 emulation window has been created and initialized.

See also: Copy and Paste methods.

Example

```
Result = VT220Client.Clear ()
If Result = False Then
    MsgBox "Unable to clear clipboard", 64, "Sample Program"
End If
```

4.3 ConfigureColor

Summary

Set background and foreground colors.

Syntax

Boolean ConfigureColor ()

Description

The foreground, background and bold colors of the VT220 terminal can be changed by calling the ConfigureColor method. This action will cause the ActiveX control to bring up a dialog box that allows the user to select new settings. The user can then choose the new colors for foreground, background and bold text from a list or select to use a custom color. For example the user may select to change the background from Black to Blue for easier viewing.

The ConfigureColor method takes no parameters and returns a boolean. If the background, foreground and bold colors can be successfully changed, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value.

Calling this method is equivalent to setting the VtAction property to VT_CONFIGURE_COLOR. This method can only be called after the VT220 emulation window has been created and initialized.

Example

```
Result = VT220Client.ConfigureColor ()
If Result = False Then
    MsgBox "Unable to configure colors", 64, "Sample Program"
End If
```

4.4 ConfigureKeyboard

Summary

Configure keyboard mappings.

Syntax

Boolean ConfigureKeyboard ()

Description

To map the keys on the keyboard, call the ConfigureKeyboard method. This action will cause the ActiveX control to bring up a dialog box that allows the user to select new settings. The dialog displays a "keyboard" and lets the user map any key, or any key in combination with the shift, ctrl or shift ctrl to a custom command or setting. For example a user can configure the keyboard to send "ls" every time shift "l" is pressed.

The ConfigureKeyboard method takes no parameters and returns a boolean. If the keyboard can be successfully mapped, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value.

Calling this method is equivalent to setting the VtAction property to VT_CONFIGURE_KEYBOARD. This method can only be called after the VT220 emulation window has been created and initialized.

Example

```
Result = VT220Client.ConfigureKeyboard ()
If Result = False Then
    MsgBox "Unable to configure the keyboard", 64, "Sample Program"
End If
```

4.5 ConfigureTerminal

Summary

Configure terminal settings.

Syntax

Boolean ConfigureTerminal ()

Description

The current VT220 terminal settings (for example, inverse video) can be changed by calling the ConfigureTerminal method. This action will cause the ActiveX control to bring up a dialog box that allows the user to select new settings. The dialog allows the user to set the type of emulation (VT220-8BIT, VT220-7BIT, VT100 and VT52) and configure the behavior of the terminal.

Note: The AutoPrint and PrintCtrl options can only be selected if there is a default printer.

The ConfigureTerminal method takes no parameters and returns a boolean. If the terminal can be successfully set, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value.

Calling this method is equivalent to setting the VtAction property to VT_CONFIGURE_TERMINAL. This method can only be called after the VT220 emulation window has been created and initialized.

Example

```
Result = VT220Client.ConfigureTerminal ()  
If Result = False Then  
    MsgBox "Unable to configure the terminal", 64, "Sample Program"  
End If
```

4.6 Connect

Summary

Establish a VT session.

Syntax

Boolean Connect (*Host*)

Host String

Description

The Connect method establishes a connection to a remote server. Before calling the Connect method, the Port property must be set to the remote port on which the remote service is running (most servers use the default remote service port 23 for Telnet).

The Connect method takes a host name (*Host*) as its parameter and returns a boolean. *Host* must be the name or internet address (in dotted decimal notation) of the remote server. If a connection is successfully established, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value.

If the connection can be established, then the OnConnect event will be fired. If the connection cannot be established, then the OnError event will be fired. These events will occur before the next statement (i.e. the statement following the call to the Connect method) is executed. The application should set a flag in the OnConnect and OnError events, so that it can determine if the session has been established or not. In addition, the application may want to display an error message in the OnError event to inform the user that the connection could not be established. Please check the reference page of the OnError event for a complete listing of error codes.

Calling this method is equivalent to setting the Action property to ACTION_CONNECT. This method can only be called after the VT220 emulation window has been created and initialized.

Example

```
Result = VT220Client.Connect ("speedy.distinct.com")
If Result = False Then
    MsgBox "Cannot establish connection", 64, "Sample Program"
End If
```

4.7 Copy

Summary

Copy highlighted text to the clipboard.

Syntax

Boolean Copy ()

Description

Clipboard activities, such as copy and paste can be performed in the emulation window. Copying the selected text in the VT220 emulation window to the clipboard can be accomplished by calling the Copy method. The Paste method can then be called to paste the copied text into the emulation window. The Clear method can be used to clear the text on the clipboard.

Before calling this method, an application must make sure that this action can be performed by verifying that the CopyClipboard property is True. This method can only be called after the VT220 emulation window has been created and initialized.

The Copy method takes no parameters and returns a boolean. If the highlighted text can be successfully copied into the buffer, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value.

Calling this method is equivalent to setting the VtAction property to VT_COPY_CLIPBOARD.

See also: CopyClipboard property, Clear and Paste methods.

Example

```
Result = VT220Client.Copy ()
If Result = False Then
    MsgBox "Unable to copy text to clipboard", 64, "Sample Program"
End If
```

4.8 CurrentLocation

Summary

Retrieve current cursor location.

Syntax

Boolean CurrentLocation ()

Description

To retrieve the current cursor position, call the CurrentLocation method. If the action was successful, the Row and Column properties will contain the current row and column position of the cursor. If not, the OnError event will be fired before the next line of code is reached.

The CurrentLocation method takes no parameters and returns a boolean. If the current cursor position can be successfully retrieved, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value.

Calling this method is equivalent to setting the VtAction property to VT_CURRENT_LOCATION. This method can only be called after the VT220 emulation window has been created and initialized.

Example

```
Result = VT220Client.CurrentLocation ()  
If Result = False Then  
    MsgBox "Unable to retrieve current cursor position", 64, "Sample Program"  
End If
```

4.9 CustomFont

Summary

Select a custom font.

Syntax

Boolean CustomFont ()

Description

To select a custom font, call the CustomFont method. This action will cause the ActiveX control to bring up a dialog box that allows the user to select new settings. The dialog allows the user to select a font type (I.e. Courier New Roman...), size (10 points) and style (Bold) to use as the default text type in the current emulation window.

The CustomFont method takes no parameters and returns a boolean. If the custom font can be successfully set, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value.

Calling this method is equivalent to setting the VtAction property to VT_SET_CUSTOM_FONT. This method can only be called after the VT220 emulation window has been created and initialized.

Example

```
Result = VT220Client.CustomFont ()  
If Result = False Then  
    MsgBox "Unable to set custom font", 64, "Sample Program"  
End If
```

4.10 DataB

Summary

Retrieve binary data.

Syntax

Long DataB (*Buffer*, *Bytes*)

<i>Buffer</i>	Variant
<i>Bytes</i>	Long

Description

The DataB method retrieves the binary data and passes it to the application.

The DataB method takes a buffer (*Buffer*) and the number of bytes to read (*Bytes*) as its parameters and returns the actual number of bytes retrieved. The *Buffer* parameter should be an array of bytes data type. The *Bytes* parameter specifies how many bytes should be retrieved.

If there are less data to be read than the requested *Bytes*, only the available data are read. This is reflected in the return value. If the returned value is less than the requested *Bytes*, this means there are less data available than requested. The application should also check to make sure what is the actual number of bytes read.

Example

```
Length = VT220Client.DataB (Buffer, Bytes)
```

4.11 DataCapture

Summary

Use to start and stop data capture.

Syntax

Boolean DataCapture (CaptureFile, AppendFile)

<i>CaptureFile</i>	String
<i>AppendFile</i>	Boolean

Description

To capture all incoming data to a disk file or a specified file, call the DataCapture method. This action is only legal if the CaptureData or the CaptureDataFile property is True. Capturing incoming data to a disk file or a specified file is disabled by calling this method a second time. An application can check the value of the boolean InCaptureData or InCaptureDataFile property to determine if the VT220 ActiveX control is currently capturing incoming data to a disk file or a specified file, respectively.

The DataCapture method takes a capture file name (*CaptureFile*) and an append flag (*AppendFile*) as its parameters and returns a boolean. The *CaptureFile* parameter must be set to the file name to contain the incoming data. The append flag must be set to True to append to the file and false to overwrite. A VT_CAPTURE_DATA is invoked if the file string is empty. In this case, the append boolean is ignored. This action will display a dialog box to request for the name of the destination disk file from the user. If the file string is not an empty string, then VT_CAPTURE_DATA_FILE is invoked. In this case, the append boolean is looked at whether to append or overwrite the file if it already exists.

If incoming data can be successfully captured, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value.

Calling this method is equivalent to setting the VtAction property to VT_CAPTURE_DATA or VT_CAPTURE_DATA_FILE. This method can only be called after the VT220 emulation window has been created and initialized.

Example

```
Result = VT220Client.DataCapture ("test", TRUE)
If Result = False Then
    MsgBox "Cannot capture incoming data to file", 64, "Sample Program"
End If
```

4.12 DeleteConfig

Summary

Delete a configuration.

Syntax

Boolean DeleteConfig (*ConfigurationName*)

ConfigurationName String

Description

Some of the actions and methods can be invoked before the VT220 terminal emulation window is initialized. To delete a configuration, call the DeleteConfig method and passing it a valid configuration name.

The DeleteConfig method takes a configuration name specified by the *ConfigurationName* parameter and returns a boolean. The configuration name must be set to a valid configuration. If the configuration can be successfully deleted, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value.

Calling this method is equivalent to setting the VtAction property to VT_DELETE_CONFIGURATION.

Example

```
Result = VT220Client.DeleteConfig ("abc")
If Result = False Then
    MsgBox "Cannot delete configuration", 64, "Sample Program"
End If
```

4.13 Destroy

Summary

Destroy the VT emulation window.

Syntax

Boolean Destroy ()

Description

Before quitting, the application must destroy the emulation window it created with ACTION_INITIALIZE or Initialize by calling the Destroy method. This will destroy the emulation window and the application can no longer perform any VT220 ActiveX control operations except for ACTION_INITIALIZE or Initialize and listing, modifying, deleting and saving configurations (see VtAction or the corresponding methods).

The Destroy method takes no parameters and returns a boolean. If VT220 emulation window can be successfully destroyed, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value.

Calling this method is equivalent to setting the Action property to ACTION_DESTROY. This method can only be called after the VT220 emulation window has been created and initialized.

Example

```
Result = VT220Client.Destroy ()  
If Result = False Then  
    MsgBox "Unable to destroy emulation window", 64, "Sample Program"  
End If
```

4.14 Disconnect

Summary

Close the VT session.

Syntax

Boolean Disconnect ()

Description

Once a connection is no longer needed, the session can be terminated by calling the Disconnect method. An application must close all connections it has created before it quits. Before quitting, the application must destroy the emulation window it created with ACTION_INITIALIZE or Initialize.

The Disconnect method takes no parameters and returns a boolean. If session can be successfully closed, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value.

Calling this method is equivalent to setting the Action property to ACTION_DISCONNECT.

Example

```
Result = VT220Client.Disconnect ()  
If Result = False Then  
    MsgBox "Cannot close session", 64, "Sample Program"  
End If
```

4.15 GetSelectedText

Summary

The GetSelectedText method allows you to scrape the user highlighted section of the screen.

Syntax

String GetSelectedText ()

Description

The GetSelectedText method allows you to scrape the portion of text highlighted by the user into a Buffer. It takes the first highlighted text position and the last highlighted text position as the co-ordinates and returns the data between these two co-ordinates as a string.

For this method to succeed the CopyClipboard property must be set to TRUE otherwise it will return an error.

See Also

GetText and GetScreenText methods

Example

```
Dim Buffer as string
If CopyClipboard = True Then
    Buffer = Vt220.GetSelectedText ()
End If
```

4.16 GetScreenText

Summary

The GetScreenText method allows you to scrape a section of the screen and pass the data to a buffer.

Syntax

String GetScreenText (x1, y1, x2, y2)

<i>x1</i>	is the position of the first character that you want to scrape
<i>y1</i>	is the position of the first line from which screen scraping is to start.
<i>x2</i>	is the position of the last character of the text portion that you want to scrape
<i>y2</i>	is the position of the last line of the text that you want to scrape

Description

The GetScreenText method allows you to pass portions of text that is displayed in a VT220 emulation window to a buffer. This means that if you have an application which displays data in predefined positions you can easily scrape that data by calling this method with the specified coordinates for the first and last characters of the text to be captured, and pass it to your application.

For this method to succeed the CaptureScreen property must be set to TRUE.

See Also

GetText method

Example

```
Dim myvariable as string
If Vt220Client.CaptureScreen = True Then
    myvariable = Vt220.GetScreenText (1,10,8,12)
End If
```

4.17 GetText

Summary

The GetText method allows you to scrape a section of scrollbar buffer and pass the data to a buffer.

Syntax

String GetText (x1, y1, x2, y2)

x1	is the position of the first character that you want to scrape
y1	is the position of the first line from which screen scraping is to start.
x2	is the position of the last character of the text portion that you want to scrape
y2	is the position of the last line of the text that you want to scrape

Description

The GetText method allows you to pass portions of text that is currently in the scrollbar buffer to a buffer. This means that if you have an application which displays data in predefined positions you can easily scrape that data by calling this method with the specified co-ordinates for the first and last characters of the text to be captured, and pass it to your application.

For this method to succeed the CaptureScreen property must be set to TRUE.

Note

The scrollbar buffer often contains leading blank lines. You must take these into consideration when calculating the text co-ordinates.

Example

```
Dim myvariable as string
If Vt220Client.CaptureScreen = True Then
    myvariable = Vt220.GetText (1,10,8,12)
End If
```

4.18 Initialize

Summary

Initialize the VT emulation window.

Syntax

Boolean Initialize (*ConfigurationName*)

ConfigurationName String

Description

The VT220 emulation must be initialized before any other action is performed (with the exception of listing, modifying, deleting and saving configurations as described in the reference page of the VtAction property or the corresponding methods). To initialize the VT220 emulation, call the Initialize method. This will create an emulation window inside the VT220 ActiveX control.

The Initialize method takes a configuration name specified by the *ConfigurationName* parameter and returns a boolean. The configuration name must be the name of a valid configuration. If the emulation window can be successfully created and initialized, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value.

Calling this method is equivalent to setting the Action property to ACTION_INITIALIZE. If the Initialize method or ACTION_INITIALIZE is called and successfully executed within an application, then the Destroy method or ACTION_DESTROY must be called before the application quits.

Example

```
Result = VT220Client.Initialize ("default")
If Result = False Then
    MsgBox "Cannot initialize emulation window", 64, "Sample Program"
End If
```

4.19 ListConfig

Summary

List all saved configurations.

Syntax

Boolean ListConfig ()

Description

Some of the actions and methods can be invoked before the VT220 terminal emulation window is initialized. A list of the names of all available configurations can be obtained by calling the ListConfig method. The list of configuration names will be delivered to the application in one or more OnList events.

The ListConfig method takes no parameters and returns a boolean. It lists all the saved configurations. If the configuration can be successfully listed, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value.

Calling this method is equivalent to setting the VtAction property to VT_LIST_CONFIGURATION.

Example

```
Result = VT220Client.ListConfig ()  
If Result = False Then  
    MsgBox "Cannot list configuration", 64, "Sample Program"  
End If
```

4.20 ModifyConfig

Summary

Modify a saved configuration.

Syntax

Boolean ModifyConfig (*ConfigurationName*)

ConfigurationName String

Description

Some of the actions and methods can be invoked before the VT220 terminal emulation window is initialized. The application can change the terminal configuration to be used before the initialization of the emulation window by calling the ModifyConfig method.

The ModifyConfig method takes a configuration name specified by the *ConfigurationName* parameter and returns a boolean. The configuration name must be set to a valid configuration. If the configuration can be successfully modified, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value.

Calling this method is equivalent to setting the VtAction property to VT_MODIFY_CONFIGURATION.

Example

```
Result = VT220Client.ModifyConfig ("default")
If Result = False Then
    MsgBox "Cannot modify configuration", 64, "Sample Program"
End If
```

4.21 Paste

Summary

Paste text in clipboard.

Syntax

Boolean Paste ()

Description

Clipboard activities, such as copy and paste, can be performed in the emulation window. Pasting text contained in the clipboard to the VT220 emulation window is done by calling the Paste method.

Any selected text in the emulation window can be copied to the clipboard using the Copy method, and the Paste method can be used to paste the contents of the clipboard into the emulation window.

Before calling this method, an application must make sure that this action can be performed by verifying that the PasteClipboard property is True. This method can only be called after the VT220 emulation window has been created and initialized.

The Paste method takes no parameters and returns a boolean. If the text in the clipboard can be successfully pasted, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value.

Calling this method is equivalent to setting the VtAction property to VT_PASTE_CLIPBOARD.

See also: PasteClipboard property and the Copy and Clear methods.

Example

```
Result = VT220Client.Paste ()  
If Result = False Then  
    MsgBox "Unable to paste text in clipboard", 64, "Sample Program"  
End If
```

4.22 PrinterFont

Summary

Select printer fonts.

Syntax

Boolean PrinterFont ()

Description

The VtAction property and some methods can be used to configure the printer to be used with the VT220 ActiveX control. The printer font can be selected by calling the PrinterFont method.

The PrinterFont method takes no parameters and returns a boolean. If the printer font can be successfully set to a user selected font, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value.

Calling this method is equivalent to setting the VtAction property to VT_PRINTER_FONT. This method can only be called after the VT220 emulation window has been created and initialized.

Example

```
Result = VT220Client.PrinterFont ()  
If Result = False Then  
    MsgBox "Cannot set printer font", 64, "Sample Program"  
End If
```

4.23 SaveConfig

Summary

Save a configuration.

Syntax

Boolean SaveConfig (*ConfigurationName*)

ConfigurationName String

Description

Some of the actions and methods can be invoked before the VT220 terminal emulation window is initialized. To save a configuration, call the SaveConfig method.

The SaveConfig method takes a configuration name specified by the *ConfigurationName* parameter and returns a boolean. The configuration name must be set to an existing or a new configuration. If the configuration can be successfully saved, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value.

Calling this method is equivalent to setting the VtAction property to VT_SAVE_CONFIGURATION.

Example

```
Result = VT220Client.SaveConfig ("testing")
If Result = False Then
    MsgBox "Cannot save configuration", 64, "Sample Program"
End If
```

4.24 SaveTermSetting

Summary

Save the terminal settings.

Syntax

Boolean SaveTermSetting ()

Description

To save the current state of the VT220 terminal emulation window, such as font, color, terminal settings and keyboard mappings, call the SaveTermSetting method. This method saves the current state to the current configuration name being used.

The SaveTermSetting method takes no parameters and returns a boolean. If the current state can be successfully saved, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value.

Calling this method is equivalent to setting the VtAction property to VT_SAVE_SETTINGS. This method can only be called after the VT220 emulation window has been created and initialized.

Example

```
Result = VT220Client.SaveTermSetting ()  
If Result = False Then  
    MsgBox "Cannot save terminal setting", 64, "Sample Program"  
End If
```

4.25 SelectPrinter

Summary

Select a default printer.

Syntax

Boolean SelectPrinter ()

Description

The VtAction property and some methods can be used to configure the printer to be used with the VT220 ActiveX control. Calling the SelectPrinter method will select the default printer to be used.

The SelectPrinter method takes no parameters and returns a boolean. If a default printer can be successfully selected, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value.

Calling this method is equivalent to setting the VtAction property to VT_PRINTER_SELECT. This method can only be called after the VT220 emulation window has been created and initialized.

Example

```
Result = VT220Client.SelectPrinter ()  
If Result = False Then  
    MsgBox "Cannot select printer", 64, "Sample Program"  
Exit Sub  
End If
```

4.26 ScreenCapture

Summary

Capture screen data into a file.

Syntax

Boolean ScreenCapture (*CaptureFile*, *AppendFile*)

<i>CaptureFile</i>	String
<i>AppendFile</i>	Boolean

Description

To capture the data currently displayed on the screen of the VT220 emulation window to a disk file or a specified file, call the ScreenCapture method. Before doing this, an application must make sure that this action can be performed by verifying that the CaptureScreen or the CaptureScreenFile property is True.

The ScreenCapture method takes a capture file name (*CaptureFile*) and an append flag (*AppendFile*) as its parameters and returns a boolean. The *CaptureFile* parameter must be set to the file name to contain the incoming data. The append flag must be set to True to append to the file and false to overwrite. A VT_CAPTURE_SCREEN is invoked if the file string is empty. In this case, the append boolean is ignored. This action will display a dialog box to request for the name of the destination disk file from the user. If the file string is not an empty string, then VT_CAPTURE_SCREEN_FILE is invoked. In this case, the append boolean is looked at whether to append or overwrite the file if it already exists.

If the data currently displayed on the VT220 emulation window can be successfully captured, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value.

Calling this method is equivalent to setting the VtAction property to VT_CAPTURE_SCREEN or VT_CAPTURE_SCREEN_FILE. This method can only be called after the VT220 emulation window has been created and initialized.

Example

```
Result = VT220Client.ScreenCapture ("test", TRUE)
If Result = False Then
    MsgBox "Cannot capture data currently on the screen", 64, "Sample Program"
End If
```

4.27 Transparent

Summary

Transparent printer mode.

Syntax

Boolean Transparent ()

Description

The VtAction property and some methods can be used to configure the printer to be used with the VT220 ActiveX control. The printer font selected by the VT_PRINTER_FONT or PrinterFont method can be replaced with the default printer font by calling the Transparent method.

The Transparent method takes no parameters and returns a boolean. If the default printer font can be successfully set, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value.

Calling this method is equivalent to setting the VtAction property to VT_TRANSPARENT. This method can only be called after the VT220 emulation window has been created and initialized.

Example

```
Result = VT220Client.Transparent ()  
If Result = False Then  
    MsgBox "Cannot set to default printer font", 64, "Sample Program"  
End If
```

4.28 VtFont

Summary

Set the terminal emulation font.

Syntax

Boolean VtFont (*CurrentFont*)

CurrentFont Integer

Description

To select the type of font to be used for VT220 emulation, call the VtFont method and pass the font type.

The VtFont method takes a current font type (*CurrentFont*) as its parameter and returns a boolean. The current font must be set to a valid font option (refer to the CurrentFont property for a list of fonts). If the terminal emulation font can be successfully set, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value.

Calling this method is equivalent to setting the VtAction property to VT_SET_VTFONT. This method can only be called after the VT220 emulation window has been created and initialized.

The following values are valid for font type:

Value	Meaning
FONT_NONE	None
FONT_DEC	Large DEC font
FONT_SMALL	Small DEC Font
FONT_IBM	IBM Font
FONT_ANSI	ANSI Font
FONT_TERMINAL	Terminal Font
FONT_CUSTOM	Custom Font
FONT_DEC_MEDIUM	Medium DEC font

Example

```
Result = VT220Client.VtFont (FONT_DEC)
If Result = False Then
    MsgBox "Cannot set terminal font", 64, "Sample Program"
End If
```


