



# PowerTerm<sup>®</sup> WebConnect WebView

## Developer's Manual

Version 5.1.0

**Ericom North America**  
Ericom Software Inc.  
231 Herbert Ave., Bldg. #4  
Closter, NJ 07624 USA  
Tel: +1 201 767 2210  
Fax: +1 201 767 2205  
Toll Free: 1 888 769 7876  
Email: [info@ericom.com](mailto:info@ericom.com)

**Ericom Europe**  
Ericom Software (UK) Ltd.  
11a Victoria Square  
Droitwich, Worcestershire  
WR9 8DE United Kingdom  
Tel: +44 (0)1905 777 970  
Fax: + 44 (0)1905 777 972  
Email: [info@ericom.co.uk](mailto:info@ericom.co.uk)

**Ericom International**  
Ericom Software Ltd.  
8 Hamarpeh Street  
Har Hotzvim  
Jerusalem 91450 Israel  
Tel: +972 (0)2 591 1700  
Fax: +972 (0)2 571 4737  
Email: [info@ericom.com](mailto:info@ericom.com)

## Important Notice

This manual is subject to the following conditions and restrictions:

- The proprietary information belonging to Ericom<sup>®</sup> Software is supplied solely for the purpose of assisting explicitly and properly authorized users of PowerTerm<sup>®</sup> WebConnect.
- No part of its contents may be used for any other purpose, disclosed to any person or firm, or reproduced by any means, electronic and mechanical, without the express prior written permission of Ericom<sup>®</sup> Software.
- The text and graphics are for the purpose of illustration and reference only. The specifications on which they are based are subject to change without notice.
- The software described in this document is furnished under a license agreement. The software may be used or copied only in accordance with the terms of that agreement.
- Information in this document is subject to change without notice. Corporate and individual names and data used in examples herein are fictitious unless otherwise noted.

Copyright © 2000-2005 Ericom<sup>®</sup> Software

Ericom<sup>®</sup> and PowerTerm<sup>®</sup> are registered trademarks of Ericom<sup>®</sup> Software, which may be registered in certain jurisdictions.

Other company and brand, product and service names are trademarks or registered trademarks of their respective holders.

---

# Table of Contents

<b>Introduction to PowerTerm WebConnect WebView</b>	<b>4</b>
<b>System Requirements</b>	<b>5</b>
<b>CREATING A CONNECTION WITH HTML</b>	<b>6</b>
➤To create an HTML connection with the Downloader	6
➤To create an HTML connection without the Downloader	7
<b>POWERTERM WEBCONNECT WEBVIEW REFERENCE</b>	<b>8</b>
<b>The Control Interface: Properties, Methods, and Events</b>	<b>10</b>
<b>Control Interface Properties</b>	<b>12</b>
<b>Control Interface General Methods</b>	<b>18</b>
<b>Control Interface General Events</b>	<b>24</b>
<b>Setup Property Methods</b>	<b>38</b>
<b>Application Property Methods</b>	<b>59</b>
<b>Control Enumerators</b>	<b>97</b>
<b>CONNECTIONS AND USERS</b>	<b>103</b>
➤To create a host connection	103
➤To test a connection	104
<b>The Server As Gateway for Fat Client</b>	<b>105</b>
<b>EMULATION AND PROTOCOL TYPES</b>	<b>106</b>
<b>Index</b>	<b>117</b>

---

# Chapter 1

## Introduction to PowerTerm WebConnect WebView

PowerTerm WebConnect WebView with Open API is Ericom Software's browser-based, Web-to-host connectivity solution, which can be deployed immediately and also be customized. The output sent to a client's device is an OCX Control, which is embedded into an HTML Page or a Visual Basic application and programmable on the client side (using JavaScript or VBScript). PowerTerm WebConnect WebView features single point (zero client) installation and centralized management, which gives the administrator a high level of control.

A key advantage that PowerTerm WebConnect WebView has is that this advanced product uses nothing more than the robust and powerful, well-known and carefully debugged PowerTerm emulation engine, which is embedded in an HTML page or a Visual Basic application and exposes methods, events and properties to the client-side scripts. This allows any alterations and adjustments made to the original legacy application to be reflected in the resulting web application totally "on-the-fly".

PowerTerm WebConnect WebView can be used in any Web scenario or a Visual Basic application where interaction with legacy systems is required and where cost effectiveness, time to market and non-intrusiveness are important.

# System Requirements

To install the PowerTerm WebConnect WebView client you need to have administrative permissions.

The PowerTerm WebConnect WebView client runs on the following platforms:

- Internet Explorer 4.0 and higher.
- Windows 95 and higher.

# Chapter 2

## Creating a Connection with HTML

In order to be utilized, the OCX control has to be included in an HTML page. The connection and user definitions are created in the PowerTerm WebConnect Administration Tool. There you can also create a login script, map the keyboard, and customize settings, if desired. [For more information, see Appendix A: "Connections and Users"](#).

There are two ways to use the OCX: with or without the PowerTerm WebConnect Downloader. The Downloader component will facilitate the download process as well as look for latest updates etc on you WebConnect server. You can read more about the Downloader in PowerTerm WebConnect Administrator's Manual. You can also check out the WebView example on [clienturls.html](#)

➤ **To create an HTML connection with the Downloader:**

1. Add to your html for the downloader:

```
<OBJECT ID=Downloader WIDTH=0 HEIGHT=0 STYLE="DISPLAY:none"
        CODEBASE="ptdownloader.cab#Version=5,1,0,90"
        CLASSID="CLSID:7EC816D4-6FC3-4C58-A7DA-A770EE461602">
    <PARAM NAME="AutoActivate" VALUE="0">
    <PARAM NAME="Modal" VALUE="1">
    <PARAM NAME="Src" VALUE="webconnectocx.cab">
</OBJECT>
```

2. Add to run the downloader:

```
Downloader.Activate();
```

3. Add to run the OCX:

```
<OBJECT id="WebConnectOCX" classid="clsid:56DA5E7C-6A81-4ED9-  
A82C-944AC0E401C5" width="0" height="0" style="WIDTH:0;  
HEIGHT:0;">  
</OBJECT>
```

➤ **To create an HTML connection without the Downloader:**

1. Determine the location of the OCX controls on the HTML page using **Object** tags.

2. Specify the CAB file in codebase as shown below:

```
<OBJECT id="WebConnectOCX" style="LEFT: 0px; TOP: 0px;  
RIGHT:0px; BOTTOM: 0px;"  
classid=clsid: 7EC816D4-6FC3-4C58-A7DA-A770EE461602 width=0  
height=0 VIEWASTEXT>
```

```
<PARAM NAME="_ExtentX" VALUE="9525">
```

```
<PARAM NAME="_ExtentY" VALUE="6350">
```

```
</OBJECT>
```

3. In order to enable the PowerTerm WebConnect WebView you must first connect to PowerTerm WebConnect server using the **LoginToServer** method of **Setup** interface:

```
If (WebConnectOCX.Setup.LoginToServer("HOSTNAME", 4000,  
"USERNAME", "PASSWORD", true, -1, "OCX Session") == true)  
alert("LogintoServer");
```

4. After the server connection is established, select the previously defined connection by the **OpenSession** method of the **Setup** interface. PowerTerm WebConnect WebView upon identifying the connection, opens the session, runs a login script, and applies the keyboard mapping and all the visual settings.

# Chapter 3

## PowerTerm WebConnect WebView Reference

This section will attempt to answer the following three questions:

- What are the methods at your disposal?
- How to use them?
- When to use them?

This reference includes all the methods, their parameters and return values, and what these methods accomplish.

**N**OTE All examples appearing in this document are written in Visual Basic/VBScript or JavaScript.

Prior to any method invocation, you have to declare a reference to the OCX object in the following manner:

- **Visual Basic:**  
`Dim ObjSetup As New WebConnectOCX.Setup`
- **VBScript:**

```
Dim ObjSetup  
Set ObjSetup = CreateObject("WebConnectOCX.Setup")
```

- **Java:**

```
ObjSetup = New WebConnectOCX.Setup
```

These PowerTerm WebConnect WebView interfaces are comprised of methods described in this section and are divided into the following three categories:

- OCX methods, events, and properties
- iSetup property's methods
- iApplication property's methods

# The Control Interface: Properties, Methods, and Events

The Control interface includes the following **properties**:

- 1 [Setup property](#)
- 2 [Application property](#)
- 3 [Browser property](#)
- 4 [Left property](#)
- 5 [Top property](#)
- 6 [CurrentURL property](#)

The Control interface includes the following **methods**:

- 1 [SetFocus method](#)
- 2 [AboutBox method](#)
- 3 [SetWindowSize method](#)
- 4 [GetWindowSize method](#)
- 5 [ShowWindow method](#)
- 6 [OpenColorDlg method](#)
- 7 [OpenKeyMappingDlg method](#)
- 8 [OpenPowerPadDlg method](#)
- 9 [OpenPrinterSetupDlg method](#)
- 10 [RunLPDClient method](#)
- 11 [RunFTPClient method](#)
- 12 [CallSupport method](#)
- 13 [SendMail](#)

The Control interface includes the following **events**:

- 1 [KeyDown event](#)
- 2 [KeyUp event](#)

- 3**    [KeyPress event](#)
- 4**    [MouseDown event](#)
- 5**    [MouseUp event](#)
- 6**    [MouseMove event](#)
- 7**    [Click event](#)
- 8**    [DbClick event](#)
- 9**    [OnConnectToServerBegin event](#)
- 10**   [OnConnectToServerEnd event](#)
- 11**   [OnDisconnectFromServer event](#)
- 12**   [OnOpenSessionBegin event](#)
- 13**   [OnOpenSessionEnd event](#)
- 14**   [OnCloseSession event](#)
- 15**   [OnBlockEnd event](#)
- 16**   [OnSystemEnd event](#)
- 17**   [OnRecordEnd event](#)

---

# Control Interface Properties

## Setup

Returns an object representing the Setup interface.

The Setup interface includes the following methods:

- 1 [SetSSLCertificateFile](#)
- 2 [SetSSLCertificatePath](#)
- 3 [SetSSLAnonymous](#)
- 4 [LoginToServer method](#)
- 5 [LoginToServerConnection method](#)
- 6 [LoginDialog method](#)
- 7 [LoginDialogConnection method](#)
- 8 [OsLogin method](#)
- 9 [OsLoginConnection method](#)
- 10 [MachineLogin method](#)
- 11 [MachineLoginConnection method](#)
- 12 [IPLogin method](#)
- 13 [IPLoginConnection method](#)
- 14 [OpenSession method](#)
- 15 [CloseSession method](#)
- 16 [IsCommunicationOpen method](#)
- 17 [ShowClientToServer method](#)
- 18 [HostPublisherConnect method](#)

[Syntax](#)

**Object.Setup**

[Arguments](#)

N/A

[Return Value](#)

N/A

[Example](#)

```
Dim SetupObj  
SetupObj Set = CreateObject("WebConnectOCX.Setup")
```

See Also

[iSetup interface](#)

[iApplication interface](#)

[Application property](#)

## **Application**

Returns an object representing the Application interface.

The Application interface includes the following methods:

- 1 [ClearScreen method](#)
- 2 [SetRts method](#)
- 3 [SetDtr method](#)
- 4 [UseEmulatorAltKeys method](#)
- 5 [UseEmulatorCtrlKeys method](#)
- 6 [UseEmulatorFuncKeys method](#)
- 7 [Exec method](#)
- 8 [GetEnvironmentVar method](#)
- 9 [GetAppVar method](#)
- 10 [GetScreenText method](#)
- 11 [GetRectText method](#)
- 12 [RingBell method](#)
- 13 [GetPrinterName method](#)
- 14 [SetPrinterName method](#)
- 15 [GetPrinterFileName method](#)
- 16 [SetPrinterFileName method](#)
- 17 [PrintScreen method](#)
- 18 [PrintFile method](#)
- 19 [GetPrintDevice method](#)
- 20 [SetPrintDevice method](#)
- 21 [GetPrintScreenConvert method](#)

- 22 [SetPrintScreenConvert method](#)
- 23 [StartAutoPrint mehod](#)
- 24 [StopAutoPrint mehod](#)
- 25 [ToggleAutoPrint mehod](#)
- 26 [LockColumns method](#)
- 27 [UnlockColumns method](#)
- 28 [Display method](#)
- 29 [Message method](#)
- 30 [InputTrace method](#)
- 31 [SendRawText method](#)
- 32 [Send method](#)
- 33 [SendBreak method](#)
- 34 [Sleep method](#)
- 35 [WaitForSystem method](#)
- 36 [WaitForRecord record](#)
- 37 [WaitForBlock method](#)
- 38 [WaitForText method](#)
- 39 [WaitForTextOnScreen method](#)
- 40 [WaitForCursor method](#)
- 41 [MapKeyToDefault method](#)
- 42 [MapKeyToNull method](#)
- 43 [MapKeyToVtKey method](#)
- 44 [MapKeyToCommand method](#)
- 45 [MapKeyToScript method](#)
- 46 [SetNewCodeData method](#)
- 47 [GetCursorPos method](#)
- 48 [SetCursorPos method](#)
- 49 [SetPowerGui method](#)
- 50 [ShowHistoryScrollBar method](#)
- 51 [CopyToFile method](#)
- 52 [CopyToBitmap method](#)

- 53 [GetColorText method](#)
- 54 [SetColorText method](#)
- 55 [GetColorBackground method](#)
- 56 [SetColorBackground method](#)
- 57 [SetCommonColors method](#)
- 58 [GetPaletteColor method](#)
- 59 [GetVTButtonAttributes method](#)
- 60 [SetVTButtonAttributes method](#)
- 61 [GetVTEditAttributes method](#)
- 62 [SetVTEditAttributes method](#)
- 63 [SetFont method](#)
- 64 [RunScriptCommand method](#)
- 65 [RunScriptFile method](#)
- 66 [UpdateScriptRecording method](#)
- 67 [GetScriptRecording method](#)

#### Syntax

#### **Object.Application**

#### Arguments

N/A

#### Return Value

N/A

#### Example

```
Dim AppObj  
Set AppObj = CreateObject("WebConnectOCX.Application")
```

#### See Also

[iSetup property](#)

#### **Browser**

Returns an object representing the IWebBrowser2 interface.

#### Syntax

#### **Object.Browser**

### Arguments

N/A

### Return Value

N/A

### Remarks

The IWebBrowser2 interface provides methods to control the Microsoft Internet Explorer. More information about this interface, its methods and properties can be found in the MSDN Library.

### Left

Returns or designates the left position of PowerTerm WebConnect WebView window.

### Syntax

#### **Object.Left**

### Arguments

N/A

### Return Value

N/A

### Example

```
WebConnectOCX.Left = 20
```

### See Also

### [Top property](#)

### Top

Returns or designates the top position of PowerTerm WebConnect WebView window.

### Syntax

#### **Object.Top**

### Arguments

N/A

### Return Value

N/A

### Example

```
WebConnectOCX.Top = 20
```

### See Also

[Left property](#)

## **CurrentURL**

Contains the designated URL.

### Syntax

#### **Object.CurrentURL**

### Arguments

N/A

### Return Value

N/A

### Remarks

During an Internet Explorer WebView session, when there is an attempt to open an additional window, it will arrive at the designated URL. If there is none, then a URL Window, identical to the original, will be opened.

### Example

```
WebConnectOCX.CurrentURL = window.location
```

```
WebConnectOCX.CurrentURL = "http://www.ericom.com"
```

---

# Control Interface General Methods

## SetFocus

Activates the PowerTerm WebConnect WebView window.

[Syntax](#)

**Object.SetFocus ( )**

[Arguments](#)

N/A

[Return Value](#)

N/A

## AboutBox

Displays the About Info dialog.

[Syntax](#)

**Object.AboutBox ( )**

[Arguments](#)

N/A

[Return Value](#)

N/A

## SetWindowSize

Designates the PowerTerm WebConnect WebView window size.

[Syntax](#)

**Object.SetWindowSize** (*iWidth* as Integer, *iHeight* as Integer)

[Arguments](#)

*iWidth* [in] The new width of the PowerTerm WebConnect WebView window.

*iHeight* [in] The new height of the PowerTerm WebConnect WebView window.

[Return Value](#)

N/A

### Remarks

The *iWidth* and *iHeight* parameters are specified in number of pixels.

### See Also

[GetWindowSize method](#)

## GetWindowSize

Retrieves the value of the PowerTerm WebConnect WebView window size.

### Syntax

**Object.GetWindowSize** (*iWidth* as Integer, *iHeight* as Integer)

### Arguments

<i>iWidth</i>	[out]	Returns the current width of the PowerTerm WebConnect WebView window.
<i>iHeight</i>	[out]	Returns the current height of the PowerTerm WebConnect WebView window.

### Return Value

N/A

### Remarks

The Width and Height parameters are specified in number of pixels.

### See Also

[SetWindowSize method](#)

## ShowWindow

Shows or hides the PowerTerm WebConnect WebView window.

### Syntax

**Object.ShowWindow** (*bShow* as Boolean)

### Arguments

<i>bShow</i>	[in]	Specifies whether to hide or show the window. If this parameter is TRUE, the window is shown. If the parameter is FALSE, the window is hidden.
--------------	------	--

### Return Value

N/A

### **OpenColorDlg**

Opens the Color dialog in which the user can modify the color attributes and save them on the PowerTerm WebConnect server or on the local machine.

#### [Syntax](#)

#### **Object.OpenColorDlg ( )**

#### [Arguments](#)

N/A

#### [Return Value](#)

N/A

#### [See Also](#)

[OpenKeyMappingDlg method](#)

[OpenPowerPadDlg method](#)

[OpenPrinterSetupDlg method](#)

### **OpenKeyMappingDlg**

Opens the Key Mapping dialog in which the user can customize the key mappings and save them on the PowerTerm WebConnect server or on the local machine.

#### [Syntax](#)

#### **Object.OpenKeyMappingDlg ( )**

#### [Arguments](#)

N/A

#### [Return Value](#)

N/A

#### [See Also](#)

[OpenColorDlg method](#)

[OpenPowerPadDlg method](#)

[OpenPrinterSetupDlg method](#)

### **OpenPowerPadDlg**

Opens the PowerPad dialog in which the user can designate the PowerPad buttons that can be programmed to execute customized PSL scripts. These

settings can be saved either on the PowerTerm WebConnect server or on the local machine.

[Syntax](#)

**Object.OpenPowerPadDlg ( )**

[Arguments](#)

N/A

[Return Value](#)

N/A

[See Also](#)

[OpenColorDlg method](#)

[OpenKeyMappingDlg method](#)

[OpenPrinterSetupDlg method](#)

### **OpenPrinterSetupDlg**

Opens the Printer Setup dialog in which the user can designate the desired printer and saves the settings on the local machine.

[Syntax](#)

**Object.OpenPrinterSetupDlg ( )**

[Arguments](#)

N/A

[Return Value](#)

N/A

[See Also](#)

[OpenColorDlg method](#)

[OpenKeyMappingDlg method](#)

[OpenPowerPadDlg method](#)

### **RunLPDClient**

Runs the PrintView client. On initial use the PrintView client is downloaded from the PowerTerm WebConnect server prior to it being run.

[Syntax](#)

**Object.RunLPDClient ( )**

### Arguments

N/A

### Return Value

N/A

### See Also

[RunFTPClient method](#)

[CallSupport method](#)

## RunFTPClient

Runs the FTP client. On initial use the FTP client is downloaded from the PowerTerm WebConnect server prior to it being run.

### Syntax

**Object.RunFTPClient ( )**

### Arguments

N/A

### Return Value

N/A

### See Also

[RunLPDClient method](#)

[CallSupport method](#)

## CallSupport

Notifies the administrator or tech-support that this particular user is in need of assistance and that the individual contacted can control the user session remotely.

### Syntax

**Object.CallSupport** (*bAdmin* as Boolean)

### Arguments

*bAdmin* [in] Specifies whether to notify the administrator or the tech-support person. If this parameter is TRUE, notify the administrator. If the parameter is FALSE, notify the tech-support person.

### Return Value

N/A

See Also

[RunLPDClient method](#)

[RunFTPClient method](#)

### **SendMail**

Opens a dialog box so that the user can send a message to other users that are connected to the PowerTerm WebConnect server.

Syntax

**Object.SendMail ( )**

Arguments

N/A

Return Value

N/A

See Also

[RunLPDClient method](#)

[RunFTPClient method](#)

---

# Control Interface General Events

## KeyDown

Occurs when the user presses a key in the PowerTerm WebConnect WebView window.

### Syntax

**Object\_KeyDown** (*iKeyCode* as Integer, *iShiftState* as Integer)

### Arguments

- iKeyCode* [in] Integer value specifying a key code, such as vbKeyF1 (the F1 key) or vbKeyHome (the HOME key). To specify key codes, see the Microsoft® Visual Basic® documentation.
- iKeyState* [in] Integer value specifying a bit field with bits corresponding to the SHIFT key (bit 0), the CTRL key (bit 1), and the ALT key (bit 2). These bits correspond to the values 1, 2, and 4, respectively. Some, all, or none of the bits can be set, indicating that some, all, or none of the keys are pressed.

### Return Value

N/A

### Example

```
<SCRIPT language = "JavaScript" event = "KeyDown(iKeyCode,  
iShiftState)"  
defer for = "WebConnectOCX">  
//insert script commands//  
</SCRIPT>
```

### See Also

[KeyUp event](#)

[KeyPress event](#)

## KeyUp

Occurs when the user releases a key in the PowerTerm WebConnect WebView window.

### Syntax

**Object\_KeyUp** (*iKeyCode* as Integer, *iShiftState* as Integer)

### Arguments

- iKeyCode* [in] Integer value specifying a key code, such as `vbKeyF1` (the F1 key) or `vbKeyHome` (the HOME key). To specify key codes, see the Microsoft® Visual Basic® documentation.
- iKeyState* [in] Integer value specifying a bit field with bits corresponding to the SHIFT key (bit 0), the CTRL key (bit 1), and the ALT key (bit 2). These bits correspond to the values 1, 2, and 4, respectively. Some, all, or none of the bits can be set, indicating that some, all, or none of the keys are pressed.

### Return Value

N/A

### Example

```
<SCRIPT language = "JavaScript" event =  
"KeyUp(iKeyCode, iShiftState)"  
defer for = "WebConnectOCX">  
//insert script commands//  
</SCRIPT>
```

### See Also

[KeyDown event](#)

[KeyPress event](#)

## KeyPress

Occurs when the user presses and releases a key in the PowerTerm WebConnect WebView window.

### Syntax

**Object\_KeyPress** (*iKeyAscii* as Integer)

### Arguments

- iKeyAscii* [in] An Integer value specifying a standard numeric ANSI keycode.

### Return Value

N/A

### Example

```
<SCRIPT language = "JavaScript" event =
```

```
"KeyPress(iKeyAscii) "  
defer for = "WebConnectOCX">  
//insert script commands//  
</SCRIPT>
```

[See Also](#)

[KeyDown event](#)

[KeyUp event](#)

## MouseDown

Occurs when the user presses a mouse when there is focus in the PowerTerm WebConnect WebView window.

[Syntax](#)

**Object\_MouseDown** (*iButton* as Integer, *iShiftState* as Integer, *fX* as Single, *fY* as Single)

[Arguments](#)

<i>iButton</i>	[in]	Integer value specifying a bit field with bits corresponding to the left button (bit 0), right button (bit 1), and middle button (bit 2). These bits correspond to the values 1, 2, and 4, respectively. Only one of the bits is set, indicating the button that caused the event.
<i>iShiftState</i>	[in]	Integer value specifying a bit field with bits corresponding to the SHIFT key (bit 0), the CTRL key (bit 1), and the ALT key (bit 2). These bits correspond to the values 1, 2, and 4, respectively. Some, all, or none of the bits can be set, indicating that some, all, or none of the keys are pressed.
<i>fX</i>	[in]	Single value specifying the x-coordinate of the mouse pointer relative to the upper left-hand corner of the control window.
<i>fY</i>	[in]	Single value specifying the y-coordinate of the mouse pointer relative to the upper left-hand corner of the control window.

[Return Value](#)

N/A

[Example](#)

```
<SCRIPT language = "JavaScript" event =  
"MouseDown(iButton, iShiftState, fX, fY)"  
defer for = "WebConnectOCX">  
//insert script commands//  
</SCRIPT>
```

#### See Also

[MouseUp event](#)

[MouseMove event](#)

[Click event](#)

[DbClick event](#)

### MouseUp

Occurs when the user releases a mouse when there is focus in the PowerTerm WebConnect WebView window.

#### Syntax

**Object\_MouseUp** (*iButton* as Integer, *iShiftState* as Integer, *fX* as Single, *fY* as Single)

#### Arguments

<i>iButton</i>	[in]	Integer value specifying a bit field with bits corresponding to the left button (bit 0), right button (bit 1), and middle button (bit 2). These bits correspond to the values 1, 2, and 4, respectively. Only one of the bits is set, indicating the button that caused the event.
<i>iShiftState</i>	[in]	Integer value specifying a bit field with bits corresponding to the SHIFT key (bit 0), the CTRL key (bit 1), and the ALT key (bit 2). These bits correspond to the values 1, 2, and 4, respectively. Some, all, or none of the bits can be set, indicating that some, all, or none of the keys are pressed.
<i>fX</i>	[in]	Single value specifying the x-coordinate of the mouse pointer relative to the upper left-hand corner of the control window.
<i>fY</i>	[in]	Single value specifying the y-coordinate of the mouse pointer relative to the upper left-hand corner of the

control window.

### Return Value

N/A

### Example

```
<SCRIPT language = "JavaScript" event =  
"MouseUp(iButton, iShiftState, fX, fY) "  
defer for = "WebConnectOCX">  
//insert script commands//  
</SCRIPT>
```

### See Also

[MouseDown event](#)

[MouseMove event](#)

[Click event](#)

[DbClick event](#)

## MouseMove

Occurs when the user moves the mouse pointer within the PowerTerm WebConnect WebView window.

### Syntax

**Object\_MouseMove** (*iButton* as Integer, *iShiftState* as Integer, *fX* as Single, *fY* as Single)

### Arguments

<i>iButton</i>	[in]	Integer value specifying a bit field with bits corresponding to the left button (bit 0), right button (bit 1), and middle button (bit 2). These bits correspond to the values 1, 2, and 4, respectively. Only one of the bits is set, indicating which buttons were being pressed at the time the event occurred
<i>iShiftState</i>	[in]	Integer value specifying a bit field with the least significant bits corresponding to the SHIFT key (bit 0), the CTRL key (bit 1), and the ALT key (bit 2). These bits correspond to the values 1, 2, and 4, respectively. The shift argument indicates the state of these keys. Some, all, or none of the bits can be set, indicating that some, all, or none of the keys were pressed at

---

		the time the event occurred.
<i>fX</i>	[in]	Single value specifying the x-coordinate of the mouse pointer relative to the upper left-hand corner of the control window.
<i>fY</i>	[in]	Single value specifying the y-coordinate of the mouse pointer relative to the upper left-hand corner of the control window.

#### Return Value

N/A

#### Example

```
<SCRIPT language = "JavaScript" event =  
"MouseMove(iButton, iShiftState, fX, fY)"  
defer for = "WebConnectOCX">  
//insert script commands//  
</SCRIPT>
```

#### See Also

[MouseDown event](#)

[MouseUp event](#)

[Click event](#)

[DbClick event](#)

### Click

Occurs when a user clicks the mouse when there is focus in the PowerTerm WebConnect WebView window.

#### Syntax

#### **Object\_Click ( )**

#### Arguments

N/A

#### Return Value

N/A

#### Example

```
<SCRIPT language = "JavaScript" event = "Click()"  
defer for = "WebConnectOCX">
```

```
//insert script commands//  
</SCRIPT>
```

[See Also](#)

[MouseDown event](#)

[MouseUp event](#)

[MouseMove event](#)

[DbClick event](#)

### **DbClick**

Occurs when a user double-clicks the mouse in the PowerTerm WebConnect WebView window.

[Syntax](#)

**Object\_DbClick ( )**

[Arguments](#)

N/A

[Return Value](#)

N/A

[Example](#)

```
<SCRIPT language = "JavaScript" event = "DbClick()"  
defer for = "WebConnectOCX">  
//insert script commands//  
</SCRIPT>
```

[See Also](#)

[MouseDown event](#)

[MouseUp event](#)

[MouseMove event](#)

[Click event](#)

### **OnConnectToServerBegin**

This event occurs when the PowerTerm WebConnect WebView begins to connect to the PowerTerm WebConnect Server.

[Syntax](#)

**Object\_OnConnectToServerBegin ( )****Arguments**

N/A

**Return Value**

N/A

**Example**

```
<SCRIPT language = "JavaScript" event =  
"OnConnectToServerBegin() "  
defer for = "WebConnectOCX">  
//insert script commands//  
</SCRIPT>
```

**See Also**[OnConnectToServerEnd event](#)[OnDisconnectFromServer event](#)[OnOpenSessionBegin event](#)[OnOpenSessionEnd event](#)[OnCloseSession event](#)**OnConnectToServerEnd**

This event occurs when the PowerTerm WebConnect WebView establishes a connection to the PowerTerm WebConnect Server.

**Syntax****Object\_OnConnectToServerEnd ( )****Arguments**

N/A

**Return Value**

N/A

**Example**

```
<SCRIPT language = "JavaScript" event =  
"OnConnectToServerEnd() "  
defer for = "WebConnectOCX">  
//insert script commands//
```

</SCRIPT>

See Also

[OnConnectToServerBegin event](#)

[OnDisconnectFromServer event](#)

[OnOpenSessionBegin event](#)

[OnOpenSessionEnd event](#)

[OnCloseSession event](#)

### **OnDisconnectFromServer**

This event occurs when the PowerTerm WebConnect WebView disconnects from the PowerTerm WebConnect Server.

Syntax

**Object\_OnDisconnectFromServer ( )**

Arguments

N/A

Return Value

N/A

Example

```
<SCRIPT language = "JavaScript" event =  
"OnDisconnectFromServer() "  
defer for = "WebConnectOCX">  
//insert script commands//  
</SCRIPT>
```

See Also

[OnConnectToServerBegin event](#)

[OnConnectToServerEnd event](#)

[OnOpenSessionBegin event](#)

[OnOpenSessionEnd event](#)

[OnCloseSession event](#)

### **OnOpenSessionBegin**

This event occurs when the PowerTerm WebConnect WebView begins to open a session with the host.

#### Syntax

#### **Object\_OnOpenSessionBegin ( )**

#### Arguments

N/A

#### Return Value

N/A

#### Example

```
<SCRIPT language = "JavaScript" event =  
"OnOpenSessionBegin()"  
defer for = "WebConnectOCX">  
//insert script commands//  
</SCRIPT>
```

#### See Also

[OnConnectToServerBegin event](#)

[OnConnectToServerEnd event](#)

[OnDisconnectFromServer event](#)

[OnOpenSessionEnd event](#)

[OnCloseSession event](#)

#### **OnOpenSessionEnd**

This event occurs when the PowerTerm WebConnect WebView establishes a session with the host.

#### Syntax

#### **Object\_OnOpenSessionEnd ( )**

#### Arguments

N/A

#### Return Value

N/A

#### Example

```
<SCRIPT language = "JavaScript" event =
```

```
"OnOpenSessionEnd() "  
defer for = "WebConnectOCX">  
//insert script commands//  
</SCRIPT>
```

[See Also](#)

[OnConnectToServerBegin event](#)

[OnConnectToServerEnd event](#)

[OnDisconnectFromServer event](#)

[OnOpenSessionBegin event](#)

[OnCloseSession event](#)

### **OnCloseSession**

This event occurs when the PowerTerm WebConnect WebView closes the session with the host.

[Syntax](#)

**Object\_OnCloseSession ( )**

[Arguments](#)

N/A

[Return Value](#)

N/A

[Example](#)

```
<SCRIPT language = "JavaScript" event =  
"OnCloseSession() "  
defer for = "WebConnectOCX">  
//insert script commands//  
</SCRIPT>
```

[See Also](#)

[OnConnectToServerBegin event](#)

[OnConnectToServerEnd event](#)

[OnDisconnectFromServer event](#)

[OnOpenSessionBegin event](#)

[OnOpenSessionEnd event](#)

## **OnBlockEnd**

This event occurs when the emulation finishes processing a data block from the host.

### [Syntax](#)

### **Object\_OnBlockEnd ( )**

### [Arguments](#)

N/A

### [Return Value](#)

N/A

### [Example](#)

```
<SCRIPT language = "JavaScript" event = "OnBlockEnd() "  
defer for = "WebConnectOCX">  
//insert script commands//  
</SCRIPT>
```

### [See Also](#)

[OnSystemEnd event](#)

[OnRecordEnd event](#)

[Sleep method](#)

[WaitForSystem method](#)

[WaitForRecord record](#)

[WaitForBlock method](#)

[WaitForText method](#)

[WaitForTextOnScreen method](#)

[WaitForCursor method](#)

## **OnSystemEnd**

This event occurs when the IBM emulation finishes processing a screen from the mainframe.

### [Syntax](#)

### **Object\_OnSystemEnd ( )**

### [Arguments](#)

N/A

[Return Value](#)

N/A

[Remarks](#)

IBM emulations only.

[Example](#)

```
<SCRIPT language = "JavaScript" event = "OnSystemEnd()"
defer for = "WebConnectOCX">
//insert script commands//
</SCRIPT>
```

[See Also](#)

[OnBlockEnd event](#)

[OnRecordEnd event](#)

[Sleep method](#)

[WaitForSystem method](#)

[WaitForRecord record](#)

[WaitForBlock method](#)

[WaitForText method](#)

[WaitForTextOnScreen method](#)

WaitForCursor method

## **OnRecordEnd**

(For IBM emulations only.)

This event occurs when the IBM emulation finishes processing a screen record from the mainframe.

[Syntax](#)

**Object\_OnRecordEnd ( )**

[Arguments](#)

N/A

[Return Value](#)

N/A

[Example](#)

```
<SCRIPT language = "JavaScript" event = "RecordEnd()"
defer for = "WebConnectOCX">
//insert script commands//
</SCRIPT>
```

See Also

[OnBlockEnd event](#)

[OnSystemEnd event](#)

[Sleep method](#)

[WaitForSystem method](#)

[WaitForRecord record](#)

[WaitForBlock method](#)

[WaitForText method](#)

[WaitForTextOnScreen method](#)

[WaitForCursor method](#)

---

# Setup Property Methods

## SetSSLCertificateFile

Specifies the OCX control to connect using SSL Certificate located in a certain file or list.

### Syntax

**SetSSLCertificateFile**(*files* as String, *save* as Boolean)

### Arguments

- |              |      |  |
|--------------|------|--|
| <i>files</i> | [in] | The file or list where to look for the certificate.  |
| <i>save</i>  | [in] | Specify if you want to save the certificate locally (in case it does not already exist in the file or list). |

### Return Value

N/A

### Example

```
WebConnectOCX.Setup.  
SetSSLCertificateFile("c:\certificate.cer;d:\certificate.cer",  
true);
```

## SetSSLCertificatePath

Specifies the OCX control to connect using SSL Certificate located in a certain path.

### Syntax

**SetSSLCertificatePath**(*path* as String, *save* as Boolean)

### Arguments

- |             |      |  |
|-------------|------|--|
| <i>path</i> | [in] | The path where to look for the certificate.  |
| <i>save</i> | [in] | Specify if you want to save the certificate locally (in case it does not already exist in the file or list). |

### Return Value

N/A

### Example

```
WebConnectOCX.Setup. SetSSLCertificatePath("c:\", true);
```

## SetSSLAnonymous

Specifies the OCX control not to use SSL Certificate.

#### Syntax

#### **SetSSLAnonymous()**

#### Arguments

N/A

#### Return Value

N/A

#### Example

```
WebConnectOCX.Setup.SetSSLAnonymous ();
```

### **LoginToServer**

Connects to PowerTerm WebConnect Server.

#### Syntax

**Object.Setup.LoginToServer** (*ServerAddress* as String, *ServerPort* as String, *Username* as String, *Password* as String, *bUseSsl* as Boolean, *ReconnectMode* as EnumReconnectMode, *SessionID* as String, *Advanced* as Boolean) as Boolean

#### Arguments

<i>ServerAddress</i>	[in]	The name of a host address where the PowerTerm WebConnect server is located
<i>ServerPort</i>	[in]	The port number upon which the PowerTerm WebConnect server is "listening" for the client.
<i>Username</i>	[in]	The user name, which is defined by the PowerTerm WebConnect server for a user.
<i>Password</i>	[in]	The password, which is defined by the PowerTerm WebConnect server for a user.
<i>bUseSsl</i>	[in]	Specifies whether to connect to PowerTerm WebConnect server with or without SSL. If this parameter is TRUE, the connection is secure via SSL. If the parameter is FALSE, the connection is not secure.
<i>ReconnectMode</i>	[in]	Specifies the desired reconnect mode
<i>SessionID</i>	[in]	Specifies the name of the session that the client opens.

*Advanced* [optional] Specifies to display an extended Login dialog, including the Domain field.

#### Return Value

*True* Indicates that the Login was successful.

*False* Indicates that the Login failed.

#### Remarks

This method must be first implemented to establish a connection with the PowerTerm WebConnect server.

For further information, refer to the online documentation for the PowerTerm WebConnect Server and PowerTerm WebConnect Administration Tool.

#### Example

```
If (WebConnectOCX.Setup.LoginToServer("HOSTNAME", 4000,
"USERNAME", "PASSWORD", false, -1, "OCX Session") == true)
    alert ("Login was OK");
else
    alert ("Login was abnormal");
```

#### See Also

[LoginToServerConnection method](#)

[LoginDialog method](#)

[LoginDialogConnection method](#)

[OsLogin method](#)

[OsLoginConnection method](#)

[MachineLogin method](#)

[MachineLoginConnection method](#)

[IPLogin](#)

[IPLoginConnection method](#)

[OpenSession method](#)

[CloseSession method](#)

[IsCommunicationOpen method](#)

## **LoginToServerConnection**

---

Connects to PowerTerm WebConnect Server and opens a new connection to a host, based on the PowerTerm WebConnect connection properties.

#### Syntax

**Object.Setup.LoginToServerConnection** (*ServerAddress* as String, *ServerPort* as String, *Username* as String, *Password* as String, *ConnectionName* as String, *bUseSsl* as Integer, *ReconnectMode* as EnumReconnectMode, *SessionID* as String, *Advanced* as Boolean) as Boolean

#### Arguments

<i>ServerAddress</i>	[in]	The name of a host address where the PowerTerm WebConnect server is located
<i>ServerPort</i>	[in]	The port number upon which the PowerTerm WebConnect server is "listening" for the client.
<i>Username</i>	[in]	The user name, which is defined by the PowerTerm WebConnect server for a user.
<i>Password</i>	[in]	The password, which is defined by the PowerTerm WebConnect server for a user.
<i>ConnectionName</i>	[in]	The name of the connection, which is defined by the PowerTerm WebConnect server for a user.
<i>bUseSsl</i>	[in]	Specifies whether to connect to PowerTerm WebConnect server with or without SSL. If this parameter is TRUE, the connection is secure via SSL. If the parameter is FALSE, the connection is not secure.
<i>ReconnectMode</i>	[in]	Specifies the desired reconnect mode
<i>SessionID</i>	[in]	Specifies the name of the session that the client opens.
<i>Advanced</i>	[optional]	Specifies to display an extended Login dialog, including the Domain field.

#### Return Value

*True* Indicates that the Login was successful.

*False* Indicates that the Login failed.

#### Remarks

This method must be first implemented to establish a connection with the PowerTerm WebConnect server and subsequently opens a new connection to a host, based on the PowerTerm WebConnect connection properties.

For further information, refer to the online documentation for the PowerTerm WebConnect Server and PowerTerm WebConnect Administration Tool.

#### Example

```
If (WebConnectOCX.Setup.LoginToServerConnection
("HOSTNAME", 4000, "USERNAME", "PASSWORD"
"CONNECTIONNAME", true, 2, "OCX Session") == true)
    alert ("Login was OK");
else
    alert ("Login was abnormal");
```

#### See Also

[LoginToServer method](#)

[LoginDialog method](#)

[LoginDialogConnection method](#)

[OsLogin method](#)

[OsLoginConnection method](#)

[MachineLogin method](#)

[MachineLoginConnection method](#)

[IPLogin](#)

[IPLoginConnection method](#)

[OpenSession method](#)

[CloseSession method](#)

[IsCommunicationOpen method](#)

### LoginDialog

Opens a Login dialog for Username and Password, which upon successful completion connects to the PowerTerm WebConnect Server.

#### Syntax

**Object.Setup.LoginDialog** (*ServerAddress* as String, *ServerPort* as String, *bUseSsl* as Boolean, *ReconnectMode* as EnumReconnectMode, *SessionID* as String, *Advanced* as Boolean) as Boolean

### Arguments

<i>ServerAddress</i>	[in]	The name of a host address where the PowerTerm WebConnect server is located.
<i>ServerPort</i>	[in]	The port number upon which the PowerTerm WebConnect server is "listening" for the client.
<i>bUseSsl</i>	[in]	Specifies whether to connect to PowerTerm WebConnect server with or without SSL. If this parameter is TRUE, the connection is secure via SSL. If the parameter is FALSE, the connection is not secure.
<i>ReconnectMode</i>	[in]	Specifies the desired reconnect mode
<i>SessionID</i>	[in]	Specifies the name of the session that the client opens.
<i>Advanced</i>	[optional]	Specifies to display an extended Login dialog, including the Domain field.

### Return Value

*True* Indicates that the Login was successful.

*False* Indicates that the Login failed.

### Remarks

This method must be first implemented to establish a connection with the PowerTerm WebConnect server.

For further information, refer to the online documentation for the PowerTerm WebConnect Server and PowerTerm WebConnect Administration Tool.

### Example

```
If (WebConnectOCX.Setup.LoginDialog("HOSTNAME", 4000,
true, 1, "OCX Session") == true)
    alert ("Login was OK");
else
    alert ("Login was abnormal");
```

### See Also

[LoginToServer method](#)

[LoginToServerConnection method](#)

[LoginDialogConnection method](#)

[OsLogin method](#)

[OsLoginConnection method](#)

[MachineLogin method](#)

[MachineLoginConnection method](#)

[IPLogin](#)

[IPLoginConnection method](#)

[OpenSession method](#)

[CloseSession method](#)

[IsCommunicationOpen method](#)

## LoginDialogConnection

Opens a Login dialog for Username and Password, which upon successful completion connects to the PowerTerm WebConnect Server. Afterwards it opens a new connection to a host, based on the PowerTerm WebConnect connection properties.

### Syntax

**Object.Setup.LoginDialogConnection** (*ServerAddress* as String, *ServerPort* as String, *ConnectionName* as String, *bUseSsl* as Boolean, *ReconnectMode* as EnumReconnectMode, *SessionID* as String, *Advanced* as Boolean) as Boolean

### Arguments

<i>ServerAddress</i>	[in]	The name of a host address where the PowerTerm WebConnect Server is located.
<i>ServerPort</i>	[in]	The port number upon which the PowerTerm WebConnect Server is "listening" for the client.
<i>ConnectionName</i>	[in]	The name of the connection, which is defined by the PowerTerm WebConnect Server for a user.
<i>bUseSsl</i>	[in]	Specifies whether to connect to PowerTerm WebConnect Server with or without SSL. If this parameter is TRUE, the connection is secure via SSL. If the parameter is FALSE, the connection is not secure.
<i>ReconnectMode</i>	[in]	Specifies the desired reconnect mode.

---

<i>SessionID</i>	[in]	Specifies the name of the session that the client opens.
<i>Advanced</i>	[optional]	Specifies to display an extended Login dialog, including the Domain field.

#### Return Value

*True* Indicates that the Login was successful.

*False* Indicates that the Login failed.

#### Remarks

This method must be first implemented to establish a connection with the PowerTerm WebConnect Server and subsequently opens a new connection to a host, based on the PowerTerm WebConnect connection properties.

For further information, refer to the online documentation for the PowerTerm WebConnect Server and PowerTerm WebConnect Administration Tool.

#### Example

```
If (WebConnectOCX.Setup.LoginDialogConnection
("HOSTNAME", 4000, "CONNECTIONNAME", true) == true)
    alert ("Login was OK");
else
    alert ("Login was abnormal");
```

#### See Also

[LoginToServer method](#)

[LoginToServerConnection method](#)

[LoginDialog method](#)

[OsLogin method](#)

[OsLoginConnection method](#)

[MachineLogin method](#)

[MachineLoginConnection method](#)

[IPLogin](#)

[IPLoginConnection method](#)

[OpenSession method](#)

[CloseSession method](#)

[IsCommunicationOpen method](#)

## OsLogin

Connects to PowerTerm WebConnect Server using the OS account name.

### Syntax

**Object.Setup.OsLogin** (*ServerAddress* as String, *ServerPort* as String, *bUseSsl* as Boolean, *ReconnectMode* as EnumReconnectMode, *SessionID* as String, *Advanced* as Boolean) as Boolean

### Arguments

<i>ServerAddress</i>	[in]	The name of a host address where the PowerTerm WebConnect Server is located.
<i>ServerPort</i>	[in]	The port number upon which the PowerTerm WebConnect Server is "listening" for the client.
<i>bUseSsl</i>	[in]	Specifies whether to connect to PowerTerm WebConnect Server with or without SSL. If this parameter is TRUE, the connection is secure via SSL. If the parameter is FALSE, the connection is not secure.
<i>ReconnectMode</i>	[in]	Specifies the desired reconnect mode.
<i>SessionID</i>	[in]	Specifies the name of the session that the client opens.
<i>Advanced</i>	[optional]	Specifies to display an extended Login dialog, including the Domain field.

### Return Value

*True* Indicates that the Login was successful.

*False* Indicates that the Login failed.

### Remarks

This method must be first implemented to establish a connection with the PowerTerm WebConnect server.

Only for users without password.

For further information, refer to the online documentation for the PowerTerm WebConnect Server and PowerTerm WebConnect Administration Tool.

### Example

```
If (WebConnectOCX.Setup.OsLogin
("HOSTNAME", 4000, true, 0, "SessionID") == true)
    alert ("Login was OK");
```

```
else
    alert ("Login was abnormal");
```

#### See Also

[LoginToServer method](#)

[LoginToServerConnection method](#)

[LoginDialog method](#)

[LoginDialogConnection method](#)

[OsLoginConnection method](#)

[MachineLogin method](#)

[MachineLoginConnection method](#)

[IPLogin](#)

[IPLoginConnection method](#)

[OpenSession method](#)

[CloseSession method](#)

[IsCommunicationOpen method](#)

### OsLoginConnection

Connects to PowerTerm WebConnect Server using the OS account name and opens a new connection to a host, based on the PowerTerm WebConnect connection properties.

#### Syntax

**Object.Setup.OsLoginConnection** (*ServerAddress* as String, *ServerPort* as String, *ConnectionName* as String, *bUseSsl* as Boolean, *ReconnectMode* as EnumReconnectMode, *SessionID* as String, *Advanced* as Boolean) as Boolean

#### Arguments

<i>ServerAddress</i>	[in]	The name of a host address where the PowerTerm WebConnect Server is located.
<i>ServerPort</i>	[in]	The port number upon which the PowerTerm WebConnect Server is "listening" for the client.
<i>ConnectionName</i>	[in]	The name of the connection, which is defined by the PowerTerm WebConnect Server for a user.
<i>bUseSsl</i>	[in]	Specifies whether to connect to PowerTerm

---

		WebConnect Server with or without SSL. If this parameter is TRUE, the connection is secure via SSL. If the parameter is FALSE, the connection is not secure.
<i>ReconnectMode</i>	[in]	Specifies the desired reconnect mode.
<i>SessionID</i>	[in]	Specifies the name of the session that the client opens.
<i>Advanced</i>	[optional]	Specifies to display an extended Login dialog, including the Domain field.

#### Return Value

*True* Indicates that the Login was successful.

*False* Indicates that the Login failed.

#### Remarks

This method must be first implemented to establish a connection with the PowerTerm WebConnect Server and subsequently opens a new connection to a host, based on the PowerTerm WebConnect connection properties.

Only for users without password.

For further information, refer to the online documentation for the PowerTerm WebConnect Server and PowerTerm WebConnect Administration Tool.

#### Example

```
If (WebConnectOCX.Setup.OsLoginConnection
("HOSTNAME", 4000, "CONNECTIONNAME", true, 0, "SessionID") == true)
    alert ("Login was OK");
else
    alert ("Login was abnormal");
```

#### See Also

[LoginToServer method](#)

[LoginToServerConnection method](#)

[LoginDialog method](#)

[LoginDialogConnection method](#)

[OsLogin method](#)

[MachineLogin method](#)

[MachineLoginConnection method](#)

[IPLogin](#)

[IPLoginConnection method](#)

[OpenSession method](#)

[CloseSession method](#)

[IsCommunicationOpen method](#)

## MachineLogin

Connects to PowerTerm WebConnect Server using the machine name.

### Syntax

**Object.Setup.MachineLogin** (*ServerAddress* as String, *ServerPort* as String, *bUseSsl* as Boolean, *ReconnectMode* as EnumReconnectMode, *SessionID* as String, *Advanced* as Boolean) as Boolean

### Arguments

<i>ServerAddress</i>	[in]	The name of a host address where the PowerTerm WebConnect Server is located.
<i>ServerPort</i>	[in]	The port number upon which the PowerTerm WebConnect Server is "listening" for the client.
<i>bUseSsl</i>	[in]	Specifies whether to connect to PowerTerm WebConnect Server with or without SSL. If this parameter is TRUE, the connection is secure via SSL. If the parameter is FALSE, the connection is not secure.
<i>ReconnectMode</i>	[in]	Specifies the desired reconnect mode.
<i>SessionID</i>	[in]	Specifies the name of the session that the client opens.
<i>Advanced</i>	[optional]	Specifies to display an extended Login dialog, including the Domain field.

### Return Value

*True* Indicates that the Login was successful.

*False* Indicates that the Login failed.

### Remarks

This method must be first implemented to establish a connection with the PowerTerm WebConnect server.

Only for users without password.

---

For further information, refer to the online documentation for the PowerTerm WebConnect Server and PowerTerm WebConnect Administration Tool.

### Example

```
If (WebConnectOCX.Setup.MachineLogin
("HOSTNAME", 4000, true, 0, "SessionID") == true)
    alert ("Login was OK");
else
    alert ("Login was abnormal");
```

### See Also

[LoginToServer method](#)

[LoginToServerConnection method](#)

[LoginDialog method](#)

[LoginDialogConnection method](#)

[OsLogin method](#)

[OsLoginConnection method](#)

[MachineLoginConnection method](#)

[IPLogin](#)

[IPLoginConnection method](#)

[OpenSession method](#)

[CloseSession method](#)

[IsCommunicationOpen method](#)

## MachineLoginConnection

Connects to PowerTerm WebConnect using the machine name and opens a new connection to a host, based on the PowerTerm WebConnect connection properties.

### Syntax

**Object.Setup.MachineLoginConnection** (*ServerAddress* as String, *ServerPort* as String, *ConnectionName* as String, *bUseSsl* as Boolean, *ReconnectMode* as EnumReconnectMode, *SessionID* as String, *Advanced* as Boolean) as Boolean

### Arguments

<i>ServerAddress</i>	[in]	The name of a host address where the PowerTerm WebConnect Server is located.
----------------------	------	--

---

<i>ServerPort</i>	[in]	The port number upon which the PowerTerm WebConnect Server is "listening" for the client.
<i>ConnectionName</i>	[in]	The name of the connection, which is defined by the PowerTerm WebConnect Server for a user.
<i>bUseSsl</i>	[in]	Specifies whether to connect to PowerTerm WebConnect Server with or without SSL. If this parameter is TRUE, the connection is secure via SSL. If the parameter is FALSE, the connection is not secure.
<i>ReconnectMode</i>	[in]	Specifies the desired reconnect mode.
<i>SessionID</i>	[in]	Specifies the name of the session that the client opens.
<i>Advanced</i>	[optional]	Specifies to display an extended Login dialog, including the Domain field.

#### Return Value

*True* Indicates that the Login was successful.

*False* Indicates that the Login failed.

#### Remarks

This method must be first implemented to establish a connection with the PowerTerm WebConnect Server and subsequently opens a new connection to a host, based on the PowerTerm WebConnect connection properties.

Only for users without password.

For further information, refer to the online documentation for the PowerTerm WebConnect Server and PowerTerm WebConnect Administration Tool.

#### Example

```
If (WebConnectOCX.Setup.MashineLoginConnection
("HOSTNAME", 4000, "CONNECTIONNAME", true, 0, "SessionID") == true)
    alert ("Login was OK");
else
    alert ("Login was abnormal");
```

#### See Also

[LoginToServer method](#)

[LoginToServerConnection method](#)

[LoginDialog method](#)

[LoginDialogConnection method](#)

[OsLogin method](#)

[OsLoginConnection method](#)

[MachineLogin method](#)

[IPLogin](#)

[IPLoginConnection method](#)

[OpenSession method](#)

[CloseSession method](#)

[IsCommunicationOpen method](#)

## **IPLogin**

Connects to PowerTerm WebConnect Server using the IP address.

### [Syntax](#)

**Object.Setup.IPLogin** (*ServerAddress* as String, *ServerPort* as String, *bUseSsl* as Boolean, *ReconnectMode* as EnumReconnectMode, *SessionID* as String, *Advanced* as Boolean) as Boolean

### [Arguments](#)

<i>ServerAddress</i>	[in]	The name of a host address where the PowerTerm WebConnect Server is located.
<i>ServerPort</i>	[in]	The port number upon which the PowerTerm WebConnect Server is "listening" for the client.
<i>bUseSsl</i>	[in]	Specifies whether to connect to PowerTerm WebConnect Server with or without SSL. If this parameter is TRUE, the connection is secure via SSL. If the parameter is FALSE, the connection is not secure.
<i>ReconnectMode</i>	[in]	Specifies the desired reconnect mode.
<i>SessionID</i>	[in]	Specifies the name of the session that the client opens.
<i>Advanced</i>	[optional]	Specifies to display an extended Login dialog, including the Domain field.

### [Return Value](#)

*True* Indicates that the Login was successful.

*False* Indicates that the Login failed.

#### Remarks

This method must be first implemented to establish a connection with the PowerTerm WebConnect server.

Only for users without password.

For further information, refer to the online documentation for the PowerTerm WebConnect Server and PowerTerm WebConnect Administration Tool.

#### Example

```
If (WebConnectOCX.Setup.LoginDialogConnection
("HOSTNAME", 4000, "CONNECTIONNAME", true, 0, "SessionID") == true)
    alert ("Login was OK");
else
    alert ("Login was abnormal");
```

#### See Also

[LoginToServer method](#)

[LoginToServerConnection method](#)

[LoginDialog method](#)

[LoginDialogConnection method](#)

[OsLogin method](#)

[OsLoginConnection method](#)

[MachineLogin method](#)

[MachineLoginConnection method](#)

[IPLoginConnection method](#)

[OpenSession method](#)

[CloseSession method](#)

[IsCommunicationOpen method](#)

### **IPLoginConnection**

Connects to PowerTerm WebConnect Server using the IP address and opens a new connection to a host, based on the PowerTerm WebConnect connection properties.

## Syntax

**Object.Setup.IPLoginConnection** (*ServerAddress* as String, *ServerPort* s String, *ConnectionName* As String, *bUseSsl* As Boolean, *ReconnectMode* as EnumReconnectMode, *SessionID* as String, *Advanced* as Boolean) as Boolean

## Arguments

<i>ServerAddress</i>	[in]	The name of a host address where the PowerTerm WebConnect Server is located.
<i>ServerPort</i>	[in]	The port number upon which the PowerTerm WebConnect Server is "listening" for the client.
<i>ConnectionName</i>	[in]	The name of the connection, which is defined by the PowerTerm WebConnect Server for a user.
<i>bUseSsl</i>	[in]	Specifies whether to connect to PowerTerm WebConnect Server with or without SSL. If this parameter is TRUE, the connection is secure via SSL. If the parameter is FALSE, the connection is not secure.
<i>ReconnectMode</i>	[in]	Specifies the desired reconnect mode.
<i>SessionID</i>	[in]	Specifies the name of the session that the client opens.
<i>Advanced</i>	[optional]	Specifies to display an extended Login dialog, including the Domain field.

## Return Value

*True* Indicates that the Login was successful.

*False* Indicates that the Login failed.

## Remarks

This method must be first implemented to establish a connection with the PowerTerm WebConnect Server and subsequently opens a new connection to a host, based on the PowerTerm WebConnect connection properties.

Only for users without password.

For further information, refer to the online documentation for the PowerTerm WebConnect Server and PowerTerm WebConnect Administration Tool.

## Example

```
If (WebConnectOCX.Setup.LoginDialogConnection
```

```
("HOSTNAME", 4000, "CONNECTIONNAME", true, 0, "SessionID") == true)
    alert ("Login was OK");
else
    alert ("Login was abnormal");
```

#### See Also

[LoginToServer method](#)

[LoginToServerConnection method](#)

[LoginDialog method](#)

[LoginDialogConnection method](#)

[OsLogin method](#)

[OsLoginConnection method](#)

[MachineLogin method](#)

[MachineLoginConnection method](#)

[IPLogin](#)

[OpenSession method](#)

[CloseSession method](#)

[IsCommunicationOpen method](#)

## OpenSession

Opens new connection to a host, based on the PowerTerm WebConnect connection properties.

### Syntax

**Object.Setup.OpenSession** (*ConnectionName* as String) as Boolean

### Arguments

*ConnectionName* [in] The name of the connection, which is defined by the PowerTerm WebConnect server for a user.

### Return Value

*True* Connection to host has been successfully opened.

*False* Failed opening connection.

### Remarks

You must first login to PowerTerm WebConnect Server using one of the numerous Login methods.

### Example

```
if (WebConnectOCX.Setup.LoginToServer("HOSTNAME", 4000,
"USERNAME", "PASSWORD", true) == true)
{
    if (WebConnectOCX.Setup.OpenSession
("CONNECTIONNAME") == true)
        alert ("Connection is open");
}
```

### See Also

[LoginToServer method](#)

[LoginToServerConnection method](#)

[LoginDialog method](#)

[LoginDialogConnection method](#)

[OsLogin method](#)

[OsLoginConnection method](#)

[MachineLogin method](#)

[MachineLoginConnection method](#)

[IPLogin](#)

[IPLoginConnection method](#)

[OpenSession method](#)

[CloseSession method](#)

[IsCommunicationOpen method](#)

[OnConnectToServerBegin event](#)

[OnConnectToServerEnd event](#)

[OnDisconnectFromServer event](#)

[OnOpenSessionBegin event](#)

[OnOpenSessionEnd event](#)

[OnCloseSession event](#)

[OnCloseSession event](#)

### **CloseSession**

Closes the current host connection.

[Syntax](#)

**Object.Setup.CloseSession ( )**

[Arguments](#)

N/A

[Return Value](#)

N/A

[Example](#)

```
If WebConnectOCX.Setup.IsCommunicationOpen() = true  
WebConnectOCX.Setup.CloseSession();
```

[See Also](#)

[LoginToServer method](#)

[LoginToServerConnection method](#)

[LoginDialog method](#)

[LoginDialogConnection method](#)

[OpenSession method](#)

[IsCommunicationOpen method](#)

[OnConnectToServerBegin event](#)

[OnConnectToServerEnd event](#)

[OnDisconnectFromServer event](#)

[OnOpenSessionBegin event](#)

[OnOpenSessionEnd event](#)

[OnCloseSession event](#)

**IsCommunicationOpen**

Determines if the communication line is open.

[Syntax](#)

**Object.Setup.IsCommunicationOpen ( )** as Boolean

[Arguments](#)

N/A

[Return Value](#)

*True* Indicates that a connection is open

*False* Indicates that a connection is closed

[See Also](#)

[LoginToServer method](#)

[LoginToServerConnection method](#)

[LoginDialog method](#)

[LoginDialogConnection method](#)

[OpenSession method](#)

[CloseSession method](#)

[OnConnectToServerBegin event](#)

[OnConnectToServerEnd event](#)

[OnDisconnectFromServer event](#)

[OnOpenSessionBegin event](#)

[OnOpenSessionEnd event](#)

[OnCloseSession event](#)

### **ShowClientToServer**

Specifies if the Connecting to server dialog will be shown at login.

[Syntax](#)

**Object.Setup.ShowClientToServer** (*bShow* as Boolean)

[Arguments](#)

*bShow* [in] Specifies whether to hide or show the dialog. If this parameter is TRUE, the dialog is shown. If the parameter is FALSE, the dialog is hidden.

[Return Value](#)

N/A

---

# Application Property Methods

## ClearScreen

Clears emulation screen.

[Syntax](#)

**Object.Application.ClearScreen ( )**

[Arguments](#)

N/A

[Return Value](#)

N/A

## SetRts

Designates or clears the Ready To Send signal for COM connection.

[Syntax](#)

**Object.Application.SetRts** (*bSet* as Boolean)

[Arguments](#)

*bSet* [in] Specifies whether to set or clear the Ready To Send signal for COM connection. If this parameter is TRUE, the signal is set. If the parameter is FALSE, the signal is cleared.

[Return Value](#)

N/A

[See Also](#)

[SetDtr method](#)

## SetDtr

Designates or clears the Data Terminal Ready signal for COM connection.

[Syntax](#)

**Object.Application.SetDtr** (*bSet* as Boolean)

[Arguments](#)

*bSet* [in] Specifies whether to set or clear the Data Terminal Ready signal for COM connection. If this parameter is TRUE, the signal is set. If the parameter is FALSE, the signal is cleared.

### Return Value

N/A

### See Also

[SetRts method](#)

## UseEmulatorAltKeys

Allows or prevents the emulation to make an <Alt> key perform the operation assigned to it.

### Syntax

**Object.Application.UseEmulatorAltKeys** (*bUse* as Boolean)

### Arguments

*bUse* [in] Specifies whether to allow or to prevent the emulation to make an <Alt> key perform the operation assigned to it. If this parameter is TRUE, the emulation designated <Alt> keys are used. If the parameter is FALSE, the emulation ignores the <Alt> keys.

### Return Value

N/A

### Remarks

Any external program, which is functioning as a container to run the OCX control within it, will not handle any <Alt> keys when *bUse* is True. For example: Visual Basic or Internet Explorer.

### See Also

[UseEmulatorCtrlKeys method](#)

[UseEmulatorFuncKeys method](#)

## UseEmulatorCtrlKeys

Allows or prevents the emulation to make a <Ctrl> key perform the operation assigned to it.

### Syntax

**Object.Application.UseEmulatorCtrlKeys** (*bUse* as Boolean)

### Arguments

*bUse* [in] Specifies whether to allow or to prevent the emulation to

make an <Ctrl> key perform the operation assigned to it. If this parameter is TRUE, the emulation designated <Ctrl> keys are used. If the parameter is FALSE, the emulation ignores the <Ctrl> keys.

#### Return Value

N/A

#### Remarks

Any external program, which is functioning as a container to run the OCX control within it, will not handle any <Ctrl> keys when *bUse* is True. For example: Visual Basic or Internet Explorer.

#### See Also

[UseEmulatorAltKeys method](#)

[UseEmulatorFuncKeys method](#)

### **UseEmulatorFuncKeys**

Allows or prevents the emulation to make a <Func> key perform the operation assigned to it.

#### Syntax

**Object.Application.UseEmulatorFuncKeys** (*bUse* as Boolean)

#### Arguments

*bUse* [in] Specifies whether to allow or to prevent the emulation to make an <Func> key perform the operation assigned to it. If this parameter is TRUE, the emulation designated <Func> keys are used. If the parameter is FALSE, the emulation ignores the <Func> keys.

#### Return Value

N/A

#### Remarks

Any external program, which is functioning as a container to run the OCX control within it, will not handle any <Func> keys when *bUse* is True. For example: Visual Basic or Internet Explorer.

#### See Also

[UseEmulatorAltKeys method](#)

[UseEmulatorCtrlKeys method](#)

## Exec

Executes the specified application.

### Syntax

**Object.Application.Exec** (*ProgramName* as String, *ProgramArguments* as String) as Boolean

### Arguments

<i>ProgramName</i>	[in]	Contains the file name for the application to be executed. If the name of the executable file in this parameter does not contain a directory path, the system first searches for the executable file in the work directory and afterwards in the PATH sequence.
<i>ProgramArguments</i>	[optional]	Contains the optional parameters for the application to be executed.

### Return Value

*True* Indicates that the program was successfully executed.

*False* Indicates that the execution of the program failed.

### Example

Activates the Word program with the report.doc parameter:

```
Dim ExecName As String
Dim ExecParam As String
ExecName = "C:\Microsoft Office\Office\Winword.exe"
ExecParam = "report.doc"
If Object.Application.Exec( ExecName, ExecParam ) =
False Then
    MsgBox "Execution was failed"
End If
```

## GetEnvironmentVar

Retrieves the value of the specified environment variable of the system.

### Syntax

---

**Object.Application.GetEnvironmentVar** (*EnvironmentVar* as String) as String

**Arguments**

*EnvironmentVar* [in] The string that specifies the environment variable.

**Return Value**

*String* Specifies the value, which represents the environment variable.

**Example**

Retrieves the value of the "Lib" variable:

```
Dim EnvName, EnvVriable As String
EnvName = "Lib"
EnvVriable = Object.Application.GetEnv EnvName
MsgBox EnvVriable
```

**See Also**

[GetAppVar method](#)

**GetAppVar**

Retrieves the value from the PowerTerm WebConnect server.

**Syntax**

**Object.Application.GetAppVar** (*AppVar* as String) as String

**Arguments**

*AppVar* [in] The string that specifies the PowerTerm WebConnect Server variable.

**Return Value**

*String* Specifies the value, which represents the server variable.

**See Also**

[GetEnvironmentVar method](#)

**GetScreenText**

Returns the text from a defined area on the emulation screen.

**Syntax**

**Object.Application.GetScreenText** (*iStartCol* as Integer, *iStartRow* as Integer, *iEndCol* as Integer, *iEndRow* as Integer) as String

### Arguments

<i>iStartCol</i>	[in]	The leftmost column of the required text area of the control window.
<i>iStartRow</i>	[in]	The uppermost row of the required text area of the control window.
<i>iEndCol</i>	[in]	The rightmost column of the required text area of the control window.
<i>iEndRow</i>	[in]	The lowermost row of the required text area of the control window.

### Return Value

*String* A text that appeared on the specified area of the control window.

### Remarks

The first row must be 1.

The first column must be 1.

### Example

```
Dim ScreenText As String
ScreenText = Object.Application.GetScreenText(1,1,24,80)
MsgBox ScreenText
```

### See Also

[GetRectText method](#)

## GetRectText

Returns the text found in the specified screen rectangle.

### Syntax

**Object.Application.GetRectText** (*iStartX* as Integer, *iStartY* as Integer, *iEndX* as Integer, *iEndY* as Integer) as String

### Arguments

<i>iStartX</i>	[in]	The left screen coordinate of the required text area of the control window.
<i>iStartY</i>	[in]	The top screen coordinate of the required text area of the control window.
<i>iEndX</i>	[in]	The right screen coordinate of the required text area of the control window.

*iEndY* [in] The bottom screen coordinate of the required text area of the control window.

#### Return Value

*String* A text that appeared on the specified area of the control window.

#### Example

```
Dim RectText As String
RectText = Object.Application.GetScreenText(20, 30, 70, 40)
MsgBox RectText
```

#### See Also

[GetScreenText method](#)

### **RingBell**

Rings the bell.

#### Syntax

**Object.Application.RingBell ( )**

#### Arguments

N/A

#### Return Value

N/A

### **GetPrinterName**

Returns the name of the current designated printer.

#### Syntax

**Object.Application.GetPrinterName ( )** as String

#### Arguments

N/A

#### Return Value

*String* Returns the name of the current designated printer.

#### Example

```
Dim PrinterName As String
PrinterName = Object.Application.GetPrinterName()
```

MsgBox PrinterName

See Also

[SetPrinterName method](#)

[GetPrinterFileName method](#)

[SetPrinterFileName method](#)

[GetPrintDevice method](#)

[SetPrintDevice method](#)

[GetPrintScreenConvert method](#)

[SetPrintScreenConvert method](#)

### **SetPrinterName**

Sets the name of the current designated printer.

Syntax

**Object.Application.SetPrinterName** (*PrinterName* as String)

Arguments

*PrinterName* [in] Specifies the name of the printer you selected to designate.

Return Value

N/A

Example

```
Dim PrinterName As String
PrinterName = "\\PrinterServer//PrinterName"
Object.Application.SetPrinterName (PrinterName)
```

See Also

[GetPrinterName method](#)

[GetPrinterFileName method](#)

[SetPrinterFileName method](#)

[GetPrintDevice method](#)

[SetPrintDevice method](#)

[GetPrintScreenConvert method](#)

[SetPrintScreenConvert method](#)

### **GetPrinterFileName**

Returns the name of the designated print output (file or device name).

#### [Syntax](#)

**Object.Application.GetPrinterFileName ( )** as String

#### [Arguments](#)

N/A

#### [Return Value](#)

*String* Returns the name of the designated print output.

#### [Example](#)

```
Dim PrinterFile As String
PrinterFile = Object.Application.GetPrinterFileName ()
MsgBox PrinterFile
```

#### [See Also](#)

[GetPrinterName method](#)

[SetPrinterName method](#)

[SetPrinterFileName method](#)

[GetPrintDevice method](#)

[SetPrintDevice method](#)

[GetPrintScreenConvert method](#)

[SetPrintScreenConvert method](#)

### **SetPrinterFileName**

Sets the name of the designated print output (file or device name).

#### [Syntax](#)

**Object.Application.SetPrinterFileName** (*PrinterFileName* as String)

#### [Arguments](#)

*PrinterFileName* [in] Specifies the name of the print output you selected to designate.

#### [Return Value](#)

N/A

#### [Example](#)

```
Dim PrinterFile As String
PrinterFile = "FAX"
Object.Application.SetPrinterFileName(PrinterFile)
```

See Also

[GetPrinterName method](#)

[SetPrinterName method](#)

[GetPrinterFileName method](#)

[GetPrintDevice method](#)

[SetPrintDevice method](#)

[GetPrintScreenConvert method](#)

[SetPrintScreenConvert method](#)

## PrintScreen

Prints the data presently displayed on the emulation screen to the designated print output (printer, file or device name).

Syntax

**Object.Application.PrintScreen ( )**

Arguments

N/A

Return Value

N/A

See Also

[PrintFile method](#)

[StartAutoPrint method](#)

[StopAutoPrint method](#)

[ToggleAutoPrint method](#)

## PrintFile

Prints the specified file.

Syntax

**Object.Application.PrintFile** (*FileName* as String)

Arguments

*FileName* [in] The name of the specified file destined to be printed.

Return Value

N/A

See Also

[PrintScreen method](#)

[StartAutoPrint mehod](#)

[StopAutoPrint mehod](#)

[ToggleAutoPrint mehod](#)

### **GetPrintDevice**

Receives a printing output channel of which the possibilities are a standard Windows Print Manager in text mode, DOS device or a file.

Syntax

**Object.Application.GetPrintDevice ( )** as EnumPrintDevice

Arguments

N/A

Return Value

*EnumPrintDevice* Specifies the Device type parameter.

See Also

[GetPrinterName method](#)

[SetPrinterName method](#)

[GetPrinterFileName method](#)

[SetPrinterFileName method](#)

[SetPrintDevice method](#)

[GetPrintScreenConvert method](#)

[SetPrintScreenConvert method](#)

### **SetPrintDevice**

Designates a printing output channel of which the possibilities are a standard Windows Print Manager in text mode, DOS device or a file.

Syntax

**Object.Application.SetPrintScreen** (*PrintConvert* as EnumPrintDevice)

## Arguments

*PrintConvert* [in] Specifies the device type parameter.

## Return Value

N/A

## See Also

[GetPrinterName method](#)

[SetPrinterName method](#)

[GetPrinterFileName method](#)

[SetPrinterFileName method](#)

[GetPrintDevice method](#)

[GetPrintScreenConvert method](#)

[SetPrintScreenConvert method](#)

**GetPrintScreenConvert**

Receives the parameter that indicates which character sets (IBM or Digital) or Graphics will be converted prior to slave printing. Otherwise 'None' of these modes will be implemented.

## Syntax

**Object.Application.GetPrintScreenConvert ( )** as EnumPrintScreenConvert

## Arguments

N/A

## Return Value

*EnumPrintScreenConvert* Specifies the conversion type parameter.

## See Also

[GetPrinterName method](#)

[SetPrinterName method](#)

[GetPrinterFileName method](#)

[SetPrinterFileName method](#)

[GetPrintDevice method](#)

[SetPrintDevice method](#)

[SetPrintScreenConvert method](#)

### **SetPrintScreenConvert**

Designates the parameter that indicates which character sets (IBM or Digital) or Graphics will be converted prior to slave printing. Otherwise 'None' of these modes will be implemented.

#### Syntax

**Object.Application.SetPrintScreen** (*ScreenConvert* as EnumPrintScreenConvert)

#### Arguments

*EnumPrintScreenConvert* [in] Specifies the conversion type parameter

#### Return Value

N/A

#### See Also

[GetPrinterName method](#)

[SetPrinterName method](#)

[GetPrinterFileName method](#)

[SetPrinterFileName method](#)

[GetPrintDevice method](#)

[SetPrintDevice method](#)

[GetPrintScreenConvert method](#)

### **StartAutoPrint**

Starts accumulating incoming data (while it is displayed on the emulation screen).

#### Syntax

**Object.Application.StartAutoPrint ( )**

#### Arguments

N/A

#### Return Value

N/A

#### See Also

[PrintScreen method](#)

[PrintFile method](#)

[StopAutoPrint mehod](#)

[ToggleAutoPrint mehod](#)

### **StopAutoPrint**

Prints all the data accumulated in the printing buffer of the slave printer or in the autoprint buffer.

Syntax

**Object.Application.StopAutoPrint ( )**

Arguments

N/A

Return Value

N/A

See Also

[PrintScreen method](#)

[PrintFile method](#)

[StartAutoPrint mehod](#)

[ToggleAutoPrint mehod](#)

### **ToggleAutoPrint**

Alternates between the start and stop autoprint states.

Syntax

**Object.Application.ToggleAutoPrint ( )**

Arguments

N/A

Return Value

N/A

See Also

[PrintScreen method](#)

[PrintFile method](#)

[StartAutoPrint mehod](#)

[StopAutoPrint method](#)

### **LockColumns**

Locks the column with the specified number, preventing the user from making any modifications until he performs the UnlockColumns method.

[Syntax](#)

**Object.Application.LockColumns** (*iCol* as Integer)

[Arguments](#)

*iCol* [in] Specifies the column on the screen to be locked.

[Return Value](#)

N/A

[See Also](#)

[UnlockColumns method](#)

### **UnlockColumns**

Unlocks all locked columns, allowing for the user to make modifications.

[Syntax](#)

**Object.Application.UnlockColumns** ( )

[Arguments](#)

N/A

[Return Value](#)

N/A

[See Also](#)

[LockColumns method](#)

### **Display**

For VT emulations only.

Displays a string on the current cursor position.

[Syntax](#)

**Object.Application.Display** (*DisplayText* as String)

[Arguments](#)

*DisplayText* [in] Specifies a text that will appear on the current cursor position of the control window.

#### Return Value

N/A

#### Example

```
Dim DisplayText As String
DisplayText = "Insert First Name"
Object.Application.Display DisplayText
```

#### See Also

[Message method](#)

### **Message**

Displays a message in Message Box.

#### Syntax

**Object.Application.Message** (*MessageText* as String)

#### Arguments

*MessageText* [in] Specifies the text of message.

#### Return Value

N/A

#### See Also

[Display method](#)

### **InputTrace**

Executes capture file.

#### Syntax

**Object.Application.InputTrace** (*FileName* as String)

#### Arguments

*FileName* [in] The name of the file of the screen capture.

#### Return Value

N/A

## SendRawText

Sends the specified text to the opened host without parsing.

### Syntax

**Object.Application.SendRawText** (*SendText* as String)

### Arguments

*SendText* [in] A string of data to send to a host.

### Return Value

N/A

### Remarks

The difference between the Send and SendRawText method is that the later command takes the text, which appears in parentheses and sends it as is.

### Example

Both commands below send the same string:

```
Object.Application.SendRawData "ericom" & Chr(13)
```

```
Object.Application.Send "ericom<enter>"
```

### See Also

[Send method](#)

[SendBreak method](#)

## Send

Sends the specified text to the host.

### Syntax

**Object.Application.Send** (*SendText* as String)

### Arguments

*SendText* [in] A string of data to send to a host.

### Return Value

N/A

### Remarks

Refer to Appendix A for the key sequences that may be sent to a host.

### Examples

In this example the program sends to the host the text string "Ericom" followed by <Enter>:

```
Object.Application.Send "Ericom<enter>"
```

In this example the program sends to the host the <Ctrl C> keys sequence:

```
Object.Application.Send "^c"
```

[See Also](#)

[SendRawText method](#)

[SendBreak method](#)

### **SendBreak**

Aborts looping command on the host.

[Syntax](#)

**Object.Application.SendBreak ( )**

[Arguments](#)

N/A

[Return Value](#)

N/A

[See Also](#)

[SendRawText method](#)

[Send method](#)

### **Sleep**

Pauses the program execution for a specified number of seconds.

[Syntax](#)

**Object.Application.Sleep** (*iSeconds* as Integer)

[Arguments](#)

*iSeconds* [in] Specifies the number of seconds.

[Return Value](#)

N/A

[Example](#)

```
Object.Application.Sleep 60
```

[See Also](#)

[WaitForSystem method](#)

[WaitForRecord record](#)

[WaitForBlock method](#)

[WaitForText method](#)

[WaitForTextOnScreen method](#)

[WaitForCursor method](#)

[OnSystemEnd event](#)

[OnBlockEnd event](#)

[OnRecordEnd event](#)

## **WaitForSystem**

(For IBM emulations only.)

Waits for the IBM emulation to notify it when the emulation has finished processing a screen.

[Syntax](#)

**Object.Application.WaitForSystem** (*iSeconds* as Integer) as Boolean

[Arguments](#)

*iSeconds* [in] Determines how long PowerTerm PowerTerm WebConnect WebView will wait for a screen before the time limit is exceeded.

[Return Value](#)

*True* Indicates that a screen was successfully received.

*False* Indicates that timeout occurred.

[Remarks](#)

At the time of processing, the control window displays the "X SYSTEM" message in the status bar indicating that the PowerTerm WebConnect WebView cannot accept any commands.

[Example](#)

```
If Object.Application.WaitForSystem(10) = False Then
    MsgBox "error in connection"
End if
```

[See Also](#)

[Sleep method](#)

[WaitForRecord record](#)

[WaitForBlock method](#)

[WaitForText method](#)

[WaitForTextOnScreen method](#)

[WaitForCursor method](#)

[OnSystemEnd event](#)

[OnBlockEnd event](#)

[OnRecordEnd event](#)

## **WaitForRecord**

(For IBM emulations.)

Waits for the next screen record from the mainframe.

[Syntax](#)

**Object.Application.WaitForRecord** (*iSeconds* as Integer, *iRecords* as Integer)  
as Boolean

[Arguments](#)

*iSeconds* [in] Determines how long PowerTerm WebConnect  
WebView will wait for a record before the time limit  
is exceeded.

*iRecords* [optional] Specifies the amount of records to wait for.

[Return Value](#)

*True* Indicates that a record was successfully received.

*False* Indicates that timeout occurred.

[Example](#)

```
If Object.Application.WaitForRecord(10) = False Then  
    MsgBox "error in connection"  
End if
```

[See Also](#)

[Sleep method](#)

[WaitForSystem method](#)

[WaitForBlock method](#)

[WaitForText method](#)

[WaitForTextOnScreen method](#)

[WaitForCursor method](#)

[OnSystemEnd event](#)

[OnBlockEnd event](#)

[OnRecordEnd event](#)

## **WaitForBlock**

Waits for the end of block data.

**Syntax**

**Object.Application.WaitForBlock ( )** as Boolean

**Arguments**

N/A

**Return Value**

*True* Indicates that a block was successfully received.

*False* Indicates that timeout occurred.

**Example**

```
If Object.Application.WaitForBlockEnd() = False then
    MsgBox "error in connection"
End if
```

**See Also**

[Sleep method](#)

[WaitForSystem method](#)

[WaitForRecord record](#)

[WaitForText method](#)

[WaitForTextOnScreen method](#)

[WaitForCursor method](#)

[OnSystemEnd event](#)

[OnBlockEnd event](#)

[OnRecordEnd event](#)

## WaitForText

Waits for specific string received from the host.

### Syntax

**Object.Application.WaitForText** (*Token* as String, *iSeconds* as Integer) as Boolean

### Arguments

*Token* [in] Specifies a text to locate within a character stream.  
*iSeconds* [in] Determines how long PowerTerm WebConnect WebView will wait for the token before the time limit is exceeded.

### Return Value

*True* Indicates that the specified token occurred within the character stream.

*False* Indicates that timeout occurred.

### Example

```
If Object.Application.WaitForText("User:",30)=False Then
    MsgBox "Host is responding too slowly"
End If
```

### See Also

[Sleep method](#)

[WaitForSystem method](#)

[WaitForRecord record](#)

[WaitForBlock method](#)

[WaitForTextOnScreen method](#)

[WaitForCursor method](#)

[OnSystemEnd event](#)

[OnBlockEnd event](#)

[OnRecordEnd event](#)

## WaitForTextOnScreen

Waits for a specific string (received from the host) to appear in a specified screen area.

### Syntax

---

**Object.Application.WaitForTextOnScreen** (*Token* as String, *iSeconds* as Integer, *iCol* as Integer, *iRow* as Integer) as Boolean

#### Arguments

<i>Token</i>	[in]	Specifies a text to locate within a character stream.
<i>iSeconds</i>	[in]	Determines how long PowerTerm WebConnect WebView will wait for the token before the time limit is exceeded.
<i>iCol</i>	[in]	Specifies the column of the emulation window where the token is expected.
<i>iRow</i>	[in]	Specifies the row of the emulation window where the token is expected.

#### Return Value

*True* Indicates that the specified token occurred within the character stream.

*False* Indicates that the timeout occurred.

#### Example

```
If ObjectApplication.WaitForTextOnScreen
("User:", 30, 5, 10) = False Then
    MsgBox "Host is responding too slowly"
End If
```

#### See Also

[Sleep method](#)

[WaitForSystem method](#)

[WaitForRecord record](#)

[WaitForBlock method](#)

[WaitForText method](#)

[WaitForCursor method](#)

[OnSystemEnd event](#)

[OnBlockEnd event](#)

[OnRecordEnd event](#)

#### WaitForCursor

Waits for the cursor to arrive at a specified screen position.

#### Syntax

---

**Object.Application.WaitForCursor** (*iSeconds* as Integer, *iCol* as Integer, *iRow* as Integer) as Boolean

#### Arguments

- iSeconds* [in] Determines how long PowerTerm WebConnect WebView will wait for the cursor to arrive at the designated position before the time limit is exceeded.
- iCol* [in] Specifies the column of the emulation window where the cursor is expected.
- iRow* [in] Specifies the row of the emulation window where the cursor is expected.

#### Return Value

*True* Indicates that the specified cursor appeared in the designated position.

*False* Indicates that the timeout has exceeded.

#### Example

```
If ObjectApplication.WaitForCursorPosition  
(30, 5, 10) = False Then  
    MsgBox "Host is responding too slowly"  
End If
```

#### See Also

[Sleep method](#)

[WaitForSystem method](#)

[WaitForRecord record](#)

[WaitForBlock method](#)

[WaitForText method](#)

[WaitForTextOnScreen method](#)

[OnSystemEnd event](#)

[OnBlockEnd event](#)

[OnRecordEnd event](#)

#### MapKeyToDefault

Configures a PC key to its default.

#### Syntax

---

**Object.Application.MapKeyToDefaultPcKey** (*PcKey* as String)**Arguments**

*PcKey* [in] Specifies the PC key that is configured.

**Return Value**

N/A

**See Also**

[MapKeyToNull method](#)

[MapKeyToVtKey method](#)

[MapKeyToCommand method](#)

[MapKeyToScript method](#)

**MapKeyToNull**

Configures a PC key to be inoperable.

**Syntax**

**Object.Application.MapKeyToNull** (*PcKey* as String)

**Arguments**

*PcKey* [in] Specifies the PC key that is configured.

**Return Value**

N/A

**See Also**

[MapKeyToDefault method](#)

[MapKeyToVtKey method](#)

[MapKeyToCommand method](#)

[MapKeyToScript method](#)

**MapKeyToVtKey**

Configures a PC key to send a VT key.

**Syntax**

**Object.Application.MapKeyToVtKey** (*PcKey* as String, *VtKey* as String)

**Arguments**

*PcKey* [in] Specifies the PC key that is configured.

*VtKey* [in] Specifies the VT key that is configured to the specified PC key.

#### Return Value

N/A

#### See Also

[MapKeyToDefault method](#)

[MapKeyToNull method](#)

[MapKeyToCommand method](#)

[MapKeyToScript method](#)

### **MapKeyToCommand**

Configures a PC key to execute a specific PSL command.

#### Syntax

**Object.Application.MapKeyToCommand** (*PcKey* as String, *Command* as String)

#### Arguments

*PcKey* [in] Specifies the key that is being mapped.

*Command* [in] Specifies the PSL command that is configured to the specified PC key.

#### Return Value

N/A

#### Remarks

For further information on PSL commands, refer to the PowerTerm PSL Reference.

#### See Also

[MapKeyToDefault method](#)

[MapKeyToNull method](#)

[MapKeyToVtKey method](#)

[MapKeyToScript method](#)

### **MapKeyToScript**

Configures a PC key to run a PSL script file.

### Syntax

**Object.Application.MapKeyToScript** (*PcKey* as String, *ScriptFile* as String)

### Arguments

*PcKey* [in] Specifies the key that is being mapped.  
*ScriptFile* [in] Specifies the script file that is configured to the specified PC key.

### Return Value

N/A

### Remarks

For further information on PSL commands, refer to the PowerTerm PSL Reference.

### See Also

[MapKeyToDefault method](#)

[MapKeyToNull method](#)

[MapKeyToVtKey method](#)

[MapKeyToCommand method](#)

## SetNewCodeData

Determines whether to use the 3270 New Code.

### Syntax

**Object.Application.SetNewCodeData** (*bSet* as Boolean)

### Arguments

*bSet* [in] Specifies whether to use or not to use the 3270 New Code. If the parameter is TRUE, the 3270 New Code is used. If the parameter is FALSE, then the old code is used.

### Return Value

N/A

## GetCursorPos

Receives the current cursor position in screen coordinates.

### Syntax

**Object.Application.GetCursorPos** (*iRow* as Integer, *iCol* as Integer)

### Arguments

*iRow* [out] Specifies the current row of the work window.

*iCol* [out] Specifies the current column of the work window.

### Return Value

N/A

### Remarks

The cursor position is always specified in screen coordinates and is not effected by the mapping mode of the window that contains the cursor.

### Example

```
var Row;
var Col;
Object.Application.GetCursorPos (Row, Col)
```

### See Also

[SetCursorPos method](#)

## SetCursorPos

Moves the cursor to the specified screen coordinates.

### Syntax

**Object.Application.SetCursorPos** (*iRow* as Integer, *iCol* as Integer)

### Arguments

*iRow* [in] Specifies the new row of the emulation window in which you want the cursor positioned.

*iCol* [in] Specifies the new column of the emulation window in which you want the cursor positioned.

### Return Value

N/A

### Remarks

The cursor position is always specified in screen coordinates and is not effected by the mapping mode of the window that contains the cursor.

### Example

```
Object.Application.SetCursorPos (10, 15);
```

### See Also

[GetCursorPos method](#)

### **SetPowerGui**

Designates or clears the Power GUI view of the emulation screen.

[Syntax](#)

**Object.Application.SetPowerGui** (*bSet* as Boolean)

[Arguments](#)

*bSet* [in] Specifies whether to set or clear the Power GUI view of the emulation screen. If this parameter is TRUE, the emulation screen is shown with a Power GUI view. If the parameter is FALSE, the emulation screen is shown normally.

[Return Value](#)

N/A

### **ShowHistoryScrollBar**

For VT emulations only.

Determines whether to show or hide the vertical scroll bar in the emulation screen.

[Syntax](#)

**Object.Application.ShowHistoryScrollBar** (*bShow* as Boolean)

[Arguments](#)

*bShow* [in] Specifies whether to hide or show the vertical scroll bar. If this parameter is TRUE, the vertical scroll bar is shown. If the parameter is FALSE, it is hidden.

[Return Value](#)

N/A

### **CopyToFile**

Writes the selected text to a file. If you do not select any text, the entire screen is written to the file. After you click this command, PowerTerm opens a dialog box in which you can specify the file name.

[Syntax](#)

**Object.Application.CopyToFile** ( )

### Arguments

N/A

### Return Value

N/A

### See Also

[CopyToBitmap method](#)

## CopyToBitmap

Copies the text in a bitmap format to the Clipboard or to a bitmap file

### Syntax

**Object.Application.CopyToBitmap** (*bSelection* as Boolean)

### Arguments

*bSelection* [in] Specifies whether to copy a bitmap format to the Clipboard or to a bitmap file. If this parameter is TRUE, it is copied to the Clipboard. If the parameter is FALSE, it is copied to a file

### Return Value

N/A

### See Also

[CopyToFile method](#)

## GetColorText

Receives text color and its designated palette place for emulation screen attribute.

### Syntax

**Object.Application.GetColorText** (*DisplayAttribute* as EnumDisplayAttributes, *PaletteColor* as EnumColorPalette)

### Arguments

*DisplayAttribute* [in] Specifies the display's attribute.  
*PaletteColor* [out] Specifies designated palette place for the color received.

### Return Value

*Color* Specifies the color that is applied to the text.  
Note RGB decimal format required.

See Also

[SetColorText method](#)

[GetColorBackground method](#)

[SetColorBackground method](#)

[SetCommonColors method](#)

[GetPaletteColor method](#)

### **SetColorText**

Modifies new text color for emulation screen attribute.

Syntax

**Object.Application.SetColorText** (*DisplayAttribute* as EnumDisplayAttributes, *PaletteColor* as EnumColorPalette, *NewTextColor* as Integer)

Arguments

<i>DisplayAttribute</i>	[in]	Specifies the display's attribute for the color being modified.
<i>PaletteColor</i>	[in]	Specifies one of the 16 palette colors, which will be substituted for by a new color.
<i>NewTextColor</i>	[in]	Specifies the color that is applied to the text. Note RGB decimal format required.

Return Value

N/A

Example

```
WebConnectOCX.Application.SetColorText (0, 13, 12238549);
```

See Also

[GetColorText method](#)

[GetColorBackground method](#)

[SetColorBackground method](#)

[SetCommonColors method](#)

[GetPaletteColor method](#)

## GetColorBackground

Receives background color and its designated palette place for emulation screen attribute.

### Syntax

**Object.Application.GetColorBackground** (*DisplayAttribute* as EnumDisplayAttributes, *PaletteColor* as EnumColorPalette)

### Arguments

<i>DisplayAttribute</i>	[in]	Specifies the display's attribute.
<i>PaletteColor</i>	[out]	Specifies designated palette place for the color received.

### Return Value

*Integer* Specifies the color that is applied to the background.  
Note RGB decimal format required.

### See Also

[GetColorText method](#)

[SetColorText method](#)

[SetColorBackground method](#)

[SetCommonColors method](#)

[GetPaletteColor method](#)

## SetColorBackground

Modifies new background color for emulation screen attribute.

### Syntax

**Object.Application.SetColorBackground** (*DisplayAttribute* as EnumDisplayAttributes, *PaletteColor* as EnumColorPalette, *NewBackColor* as Integer)

### Arguments

<i>DisplayAttribute</i>	[in]	Specifies the display's attribute for the color being modified.
<i>PaletteColor</i>	[in]	Specifies one of the 16 palette colors, which will be substituted for by a new color.
<i>NewBackColor</i>	[in]	Specifies the color that is applied to the background.

---

Note RGB decimal format required.

#### Return Value

N/A

#### Example

Sets the colors for a VT emulation:

```
WebConnectOCX.Application.SetColorBackground (0, 13, 12238549);
```

#### See Also

[GetColorText method](#)

[SetColorText method](#)

[GetColorBackground method](#)

[SetCommonColors method](#)

[GetPaletteColor method](#)

### **SetCommonColors**

Modifies new background and text color for all emulation screen attributes.

#### Syntax

**Object.Application.SetCommonColors** (*NewTextColor* as Integer,  
*NewBackColor* as Integer)

#### Arguments

*NewTextColor* [in] Specifies the color that is applied to the text.

Note RGB decimal format required.

*NewBackColor* [in] Specifies the color that is applied to the background.

Note RGB decimal format required.

#### Return Value

N/A

#### See Also

[GetColorText method](#)

[SetColorText method](#)

[SetColorBackground method](#)

[GetColorBackground method](#)

[GetPaletteColor method](#)

### **GetPaletteColor**

Retrieves the color from the designated palette place.

#### Syntax

**Object.Application.GetPaletteColor** (*PaletteColor* as EnumColorPalette)

#### Arguments

*PaletteColor* [in] Specifies designated palette place for the color received.

#### Return Value

*Color* Specifies the color that is applied to palette place.

Note RGB decimal format required.

#### See Also

[GetColorText method](#)

[SetColorText method](#)

[SetColorBackground method](#)

[GetColorBackground method](#)

[SetCommonColors method](#)

### **GetVTButtonAttributes**

For VT emulations only.

Returns the display attribute of a field, thus enabling Power GUI to identify the field as a button.

#### Syntax

**Object.Application.GetVTButtonAttribute ( )**

#### Arguments

N/A

#### Return Value

*EnumDisplayAttribute* Specifies the display's attribute, which enables Power GUI to identify it as a button.

#### See Also

[SetVTButtonAttributes method](#)

[GetVTEditAttributes method](#)

---

[SetVTEditAttributes method](#)

### **SetVTButtonAttributes**

For VT emulations only.

Sets the display attribute of a field, thus enabling Power GUI to identify it as a button.

[Syntax](#)

**Object.Application.SetVTButtonAttributes** (*DisplayAttribute* as EnumDisplayAttributes)

[Arguments](#)

*DisplayAttribute* [in] Specifies the display's attribute, which enables Power GUI to identify it as a button.

[Return Value](#)

N/A

[See Also](#)

[GetVTButtonAttributes method](#)

[GetVTEditAttributes method](#)

[SetVTEditAttributes method](#)

### **GetVTEditAttributes**

For VT emulations only.

Returns the display attribute of a field, thus enabling Power GUI to identify it as an editable field.

[Syntax](#)

**Object.Application.GetVTEditAttributes** ( ) as EnumDisplayAttribute

[Arguments](#)

N/A

[Return Value](#)

*EnumDisplayAttribute* Specifies the display's attribute, which enables Power GUI to identify it as an editable field.

[See Also](#)

[GetVTButtonAttributes method](#)

---

[SetVTButtonAttributes method](#)

[SetVTEditAttributes method](#)

### **SetVTEditAttributes**

For VT emulations only.

Sets the display attribute of a field, thus enabling Power GUI to identify it as an editable field.

[Syntax](#)

**Object.Application.SetVTEditAttributes** (*DisplayAttribute* as EnumDisplayAttributes)

[Arguments](#)

*DisplayAttribute* [in] Specifies the display's attribute, which enables Power GUI to identify it as an editable field.

[Return Value](#)

N/A

[See Also](#)

[GetVTButtonAttributes method](#)

[SetVTButtonAttributes method](#)

[GetVTEditAttributes method](#)

### **SetFont**

Enables selection of the system or default PowerTerm fonts to be displayed in the PowerTerm window.

[Syntax](#)

**Object.Application.SetFont** (*bPowerTerm* as Boolean)

[Arguments](#)

*bPowerTerm* [in] Specifies whether to select the system or default PowerTerm fonts to be displayed in the PowerTerm window. If this parameter is TRUE, the system fonts are selected. If the parameter is FALSE, the default PowerTerm fonts are selected.

[Return Value](#)

N/A

## RunScriptCommand

Runs PSL script command.

### Syntax

**Object.Application.RunScriptCommand** (*ScriptCommand* as String) as String

### Arguments

*ScriptCommand* [in] The PSL script command that is to be run.

### Return Value

*String* Returns the return string of the command, if the command executed successfully. Otherwise returns an empty string.

### Remarks

For further information on PSL commands, refer to the PowerTerm WebConnect Administration Tool's online help.

### See Also

[RunScriptFile method](#)

## RunScriptFile

Runs PSL script file.

### Syntax

**Object.Application.RunScriptFile** (*ScriptFileName* as String) as Boolean

### Arguments

*ScriptFileName* [in] The PSL script file that is to be run.

### Return Value

*True* Indicates that a script file ran successfully.

*False* Indicates that a script file failed.

### Remarks

For further information on PSL commands, refer to the PowerTerm WebConnect Administration Tool's online help.

### See Also

[RunScriptCommand method](#)

### **UpdateScriptRecording**

Toggles the PSL Script recording from Start to Stop.

[Syntax](#)

**Object.Application.UpdateScripRecording ( )**

[Arguments](#)

N/A

[Return Value](#)

N/A

[See Also](#)

[GetScriptRecording method](#)

### **GetScriptRecording**

Returns the current script that was recorded as a string.

[Syntax](#)

**Object.Application. GetScriptRecording ( )** as String

[Arguments](#)

N/A

[Return Value](#)

*String* Represents the recorded PSL script.

[Remarks](#)

The UpdateScriptRecording method must be invoked before using the method described above.

[See Also](#)

[UpdateScriptRecording method](#)

---

# Control Enumerators

## EnumReconnectMode

An enumerated list of available Reconnect modes. The reconnect mode of a session is established at login time and never changes during the session's life. It is requested by the client and is granted by the server.

### Parameters

**eInteractiveRM** Enables the user to modify the Reconnect mode during login to PowerTerm WebConnect server.

- Value **-1**

**eNoRM** Disables reconnect to an interrupted session.

- Value **1**

**eOnDemandRM** Enables reconnect only to sessions connected through the PowerTerm WebConnect's gateway.

- Value **2**

**eWirelessRM** Enables reconnect to any session automatically. All predefined connections opened by a wireless session use the PowerTerm WebConnect's gateway.

- Value **3**

### See Also

[LoginToServer method](#)

[LoginToServerConnection method](#)

[LoginDialog method](#)

[LoginDialogConnection method](#)

[OsLogin method](#)

[OsLoginConnection method](#)

[MachineLogin method](#)

[MachineLoginConnection method](#)

[IPLogin](#)

[IPLoginConnection method](#)

### **EnumPrintDevice**

An enumerated list of available printing output channels.

[Parameters](#)

- |                      |   |
|----------------------|---|
| <b>eNone</b>         | No destination assigned. <ul style="list-style-type: none"><li>• Value <b>0</b></li></ul>   |
| <b>ePrintManager</b> | Sends to the standard Windows Print Manager, in text mode. <ul style="list-style-type: none"><li>• Value <b>1</b></li></ul>             |
| <b>eDevice</b>       | Sends to the DOS device designated by the PowerTerm WebConnect Server. <ul style="list-style-type: none"><li>• Value <b>2</b></li></ul> |
| <b>eFile</b>         | Sends to the DOS file designated by the PowerTerm WebConnect Server. <ul style="list-style-type: none"><li>• Value <b>3</b></li></ul>   |

[See Also](#)

[GetPrintDevice method](#)

[SetPrintDevice method](#)

### **EnumPrintScreenConvert**

An enumerated list of available data conversion modes for printing.

[Parameters](#)

- |                    |  |
|--------------------|--|
| <b>eNo_Convert</b> | No data conversion will take place. <ul style="list-style-type: none"><li>• Value <b>0</b></li></ul> |
| <b>eIbm</b>        | Converts data to IBM character sets for slave printing.  |

- Value **1**

**eDigital** Converts data to Digital character sets for slave printing.

- Value **2**

**eGraphics** Select the Graphics mode sends a print screen.

- Value **3**

[See Also](#)

[GetPrintScreenConvert method](#)

[SetPrintScreenConvert method](#)

## **EnumDisplayAttributes**

Sets the display attributes.

[Parameters](#)

### **VT and General emulations:**

<b>Value</b>	<b>Enumerator Name</b>
0	eNormal
1	eBlink
2	eReverse
3	eReverse_Blink
4	eUnderline
5	eUnderline_Blink
6	eUnderline_Reverse
7	eUnderline_Reverse_Blink
8	eBold
9	eBold_Blink
10	eBold_Reverse
11	eBold_Reverse_Blink
12	eBold_Underline
13	eBold_Underline_Blink

- 14 eBold\_Underline\_Reverse
- 15 eBold\_Underline\_Reverse\_Blink

**IBM 3270 emulation:**

<b>Value</b>	<b>Enumerator Name</b>
16	eUnprotocol_Normal
17	eUnprotocol_Intensed
18	eProtected_Normal
19	eProtected_Intensed
20	eBlue
21	eRed
22	ePink
23	eGreen
24	eTurquoise
25	eYellow
26	eWhite
27	eDefault
28	eFrame
29	eStatus
64	eBackground
65	ePowerGUI_Button
66	ePowerGUI_Edit

**IBM 5250 emulation:**

<b>Value</b>	<b>Enumerator Name</b>
32	e20_Green
33	e21_Green_Reverse
34	e22_White
35	e23_White_Reverse
36	e24_Green_Underscore
37	e25_Green_Underscore_Reverse

38	e26_White_Underscore
40	e28_Red
41	e29_Red_Reverse
42	e2A_Red_Blink
43	e2B_Red_Blink_Reverse
44	e2C_Red_Underscore
45	e2D_Red_Underscore_Reverse
46	e2E_Red_Underscore_Bold
48	e30_Turquoise_Column_Separator
49	e31_Turquoise_Column_Separator_Reverse
50	e32_Yellow_Column_Separator
51	e33_Yellow_Column_Separator_Reverse
52	e34_Turquoise_Column_Separator_Underscore
53	e35_Turquoise_Column_Separator_Underscore_Reverse
54	e36_Yellow_Column_Separator_Underscore
56	e38_Pink
57	e39_Pink_Reverse
58	e3A_Blue
59	e3B_Blue_Reverse
60	e3C_Pink_Underscore
61	e3D_Pink_Underscore_Reverse
62	e3E_Blue_Underscore
28	eFrame
29	eStatus
64	eBackground
65	ePowerGUI_Button
66	ePowerGUI_Edit

See Also

[GetColorText method](#)

[SetColorText method](#)

[GetColorBackground method](#)

[SetColorBackground method](#)  
[GetVTButtonAttributes method](#)  
[SetVTButtonAttributes method](#)  
[GetVTEditAttributes method](#)  
[SetVTEditAttributes method](#)

### **EnumColorPalette**

Sets the palette colors.

#### **Parameters**

<b>Value</b>	<b>Enumerator Name</b>
0	eBlack_Color
1	ePink_Color
2	eDark_Gray_Color
3	eDark_Pink_Color
4	eBlue_Color
5	eCyan_Color
6	eDark_Blue_Color
7	eDark_Cyan_Color
8	eRed_Color
9	eYellow_Color
10	eDark_Red_Color
11	eDark_Yellow_Color
12	eGreen_Color
13	eGray_Color
14	eDark_Green_Color
15	eWhite_Color

#### **See Also**

[GetColorText method](#)  
[SetColorText method](#)  
[GetColorBackground method](#)  
[SetColorBackground method](#)

# Appendix A

## Connections and Users

Connections and users have to be defined for the WebView client. This is done in the PowerTerm WebConnect Administration Tool. You can also create a login script, map the keyboard, and customize settings there.

**N**OTE For each host you need to create at least one separate connection.

➤ **To create a host connection:**

- 1** Open the PowerTerm WebConnect Administration Tool.
- 2** Select **Action | New | Connection**. The **Add Connection** dialog box is displayed.
- 3** Enter parameters and click **OK**. The new host connection is created.

Parameters:

<b>Connection Name</b>	The name given to the new connection that you are defining. Notice that names are not case-sensitive however must be unique.
<b>Display Name</b>	The non-unique name given to the server that appears in the user's menu. This is optional.
<b>Enabled</b>	Clear the check box if you want the connection temporarily disconnected.
<b>Usage Type</b>	<b>Hidden</b> , Can only be activated from a login script.

**Child**, Owned by another connection and triggered by it.

**Regular**, Normal connection.

**Owner**, A normal connection which, when closed, will automatically shut down all associated connections (child connections, connections opened by the logins script).

#### **Optional Definitions:**

- **Security** – specify **Security** level and details for your connection. For more information see *Security* in the online documentation for the PowerTerm WebConnect Administration Tool.
- **Settings** - click **Settings** to customize the Terminal Setup. For more information see *Property Pages and their Options* in the online documentation for the PowerTerm WebConnect Administration Tool.
- **Keyboard Mappings** - click **Key Mapping** to set the PC keyboard to the emulation keyboard. For more information see *Keyboard Mapping* in the online documentation for the PowerTerm WebConnect Administration Tool.
- **Power Pad** – click **Power Pad** to customize the Power Pad and the Function buttons. For more information see *Power Pad* and *Function Buttons* in the online documentation for the PowerTerm WebConnect Administration Tool.
- **Login Script** – create a Login script to be run after communication is established.

#### ➤ **To test a connection:**

- 1** Select the desired connection from the bottom section of the Administration Tool's main screen.
- 2** Right-click **Test**. The connection is established.

## The Server As Gateway for Fat Client

The WebView client, in comparison to the Java client, is sometimes dependent on the PowerTerm WebConnect server in order to connect to a host. In such a case when the WebView client cannot “see the host” (since the client and the host are located on two distinctly separate networks) then the PowerTerm WebConnect server can act as a gateway. In order to acquire this capability you must specify the network name of the predefined connection’s target (the IP address).

The PowerTerm WebConnect Server can be accessed by a client from several different connection points, as specified in the server’s INI file. Each connection point represents a logical or physical network, i.e. Intranet and Internet. The **NetworkName** attribute of each connection point specifies the name the system administrator assigns to the logical or physical network. The server keeps track about each client’s accessed connection point. This information can be examined in the **Via** column of the *Active Session’s* list view.

For information about how to create a host connection, see “[Creating a Host Connection](#)”.

# Appendix B

## Emulation and Protocol Types

Emulation types:

<b>Emulation</b>	<b>Protocol Type</b>	<b>Parameters</b>
<b>VT52, VT100, VT220-7, VT220-8, VT320-7, VT320-8, VT420-7, VT420-8, VT525-7, VT525-8, DG, SCO-ANSI, BBS-ANSI, AIXTERM, AT386, WYSE50, WYSE50+, WYSE60, WYSE370, TVI910+, TVI920, TVI925, TVI950, TVI955, HP, DG, SIEMENS, ADDS VP A2, HZ 1500</b>	<a href="#">TELNET</a>	Host Name, Terminal Name, Port Number, Keep Alive Timeout, Set Window Size, Network Name, Type, Show Certificate
	<a href="#">LAT</a>	Service Name, Password, Network Name
	<a href="#">NWLAT</a>	
	<a href="#">NSVT</a>	Host Name, Service, Network Name
	<a href="#">CTERM</a>	Node Name, Network Name

---

<b>Emulation</b>	<b>Protocol Type</b>	<b>Parameters</b>
	<a href="#">RLOGIN</a>	Host Name, Port Number, Keep Alive Timeout, Set Window Size, Network Name
<b>3270</b>	<a href="#">TN3270</a>	Host Name, LU Name, Port Number, Keep Alive Timeout, User TN3270E Protocol, Network Name, Type, Show Certificate
	<a href="#">MS SNA Server</a>	Use Available LU, LU Name, Network Name
	<a href="#">NWSAA (IPX)</a> <a href="#">NWSAA (TCPIP)</a>	Server Name, Backup, User Name, Service Name, LU Category, LU Name, Network Name
<b>5250</b>	<a href="#">TN5250</a>	Host Name, Device Name, Port Number, Keep Alive Timeout, Message Queue, Message Library, Network Name, Type, Show Certificate

<b>Emulation</b>	<b>Protocol Type</b>	<b>Parameters</b>
	<a href="#">APPC</a>	System Name, Device Name, Message Queue, Message Library, Network Name
	<a href="#">MS SNA Server</a>	System Name, Device Name, User Name, Password, Message Queue, Message Library, Network Name
<b>TANDEM 6530</b>	<a href="#">TELNET</a>	Host Name, Service, Port Number, Keep Alive Timeout, Set Window Size, Network Name, Type, Show Certificate

Protocol Types:

### **TELNET Parameters**

- Host Name** Specifies the host computer name or the host's IP address.
- Terminal Name** Specifies the terminal name.
- Port Number** Specifies the TELNET port number (default 23).

<b>Keep Alive Timeout</b>	Periodically sends a NOP (non operation) command to the host when any number greater than zero is entered in this field. For example: Enter 600 (seconds) to "keep alive" every 10 minutes.
<b>Set Window Size</b>	Tells PowerTerm to report the window size (if asked by the host) in the telnet negotiation procedure.
<b>Network Name</b>	Only for "Fat Client" (see also Server As Gateway). Specifies generic name of the network over which communication to host is being transmitted.
<b>Type</b>	Specify which SSL protocol to use in the host connection.
<b>Show Certificate</b>	Displays the Security dialog in which the user can designate which SSL Certificate behavior to implement for your host connection.

#### **RLOGIN Parameters**

<b>Host Name</b>	Specifies the host computer name or the host's IP address.
<b>Port Number</b>	Specifies the TELNET port number (default 23).
<b>Keep Alive Timeout</b>	Periodically sends a NOP (non operation) command to the host when any number greater than zero is entered in this field. For example: Enter 600 (seconds) to "keep alive" every 10 minutes.
<b>Set Window Size</b>	Tells PowerTerm to report the window size (if asked by the host) in the telnet negotiation procedure.

**Network Name** Only for "Fat Client" (see also Server As Gateway).  
Specifies generic name of the network over which communication to host is being transmitted.

### **LAT AND NWLAT Parameters**

**Service Name** Specifies the name of the service.

**Password** Supplies the password you use on the remote host system.

**Network Name** Only for "Fat Client" (see also Server As Gateway).  
Specifies generic name of the network over which communication to host is being transmitted.

### **CTERM Parameters**

**Node Name** Specifies the host computer name or the host's IP address through which the data is transferred.

**Network Name** Only for "Fat Client" (see also Server As Gateway).  
Specifies generic name of the network over which communication to host is being transmitted.

### **TN3270 Parameters**

**Host Name** Specifies the host computer name or the host's IP address.

**LU Name** Specifies LU name.

**Port Number** Specifies the TELNET port number (default 23).

**Keep Alive Timeout** Periodically sends a NOP (non operation) command to the host when any number greater than zero is entered in this field. For example: Enter 600 (seconds) to "keep alive" every 10 minutes.

**Use TN3270E Protocol** Specifies using the TN3270E communications protocol.

**Network Name** Only for "Fat Client" (see also Server As Gateway). Specifies generic name of the network over which communication to host is being transmitted.

**Type** Specify which SSL protocol to use in the host connection.

**Show Certificate** Displays the Security dialog in which the user can designate which SSL Certificate behavior to implement for your host connection.

### **MS SNA Server Parameters for 3270**

**Use Available LU** Indicates that the LU is available to the host.

**LU Name** Specifies LU name.

**Network Name** Only for "Fat Client" (see also Server As Gateway). Specifies generic name of the network over which communication to host is being transmitted.

### **NWSAA Parameters**

**Server Name** Specifies the server name. You can specify an asterisk

as the server name, and PowerTerm will connect to the appropriate NetWare for SAA server.

**Backup** Specifies the backup server name.

**User Name** Specifies the user name that will be relayed on to the host when attempting to logon.

**Service Name** Specifies the name of the service.

**LU Category** Specifies the name of the LU Category.

**LU Name** Specifies LU name.

**Network Name** Only for "Fat Client" (see also Server As Gateway). Specifies generic name of the network over which communication to host is being transmitted.

### **TN5250 Parameters**

**Host Name** Specifies the host computer name or the host's IP address.

**Device Name** When using multiple sessions, each new session can be automatically given a new name, followed by the session number. For example, if the device name is Test, the first session would be Test1, the next session Test2, and so on. For more information see Dynamic Device Names for 5250 Emulations in the online documentation for the PowerTerm WebConnect Administration Tool.

**Port Number** Specifies the TELNET port number (default 23).

- Keep Alive Timeout** Periodically sends a NOP (non operation) command to the host when any number greater than zero is entered in this field. For example: Enter 600 (seconds) to "keep alive" every 10 minutes.
- Message Queue** Only visible for Printer emulations.  
Designates to which AS/400 message queue exception messages should be sent. For example: The AS/400 may need to tell the printer to switch to another paper tray.
- Message Library** Only visible for Printer emulations.  
Specifies which library contains the message queue for exception messages.
- Network Name** Only for "Fat Client" (see also Server As Gateway).  
Specifies generic name of the network over which communication to host is being transmitted.
- Type** Specify which SSL protocol to use in the host connection.
- Show Certificate** Displays the Security dialog in which the user can designate which SSL Certificate behavior to implement for his host connection.

### **APPC Parameters**

- System Name** Specifies the name of the system.
- Device Name** When using multiple sessions, each new session can be automatically given a new name, followed by the session number. For example, if the device name is Test, the first session would be Test1, the next session Test2, and so on. For more information see Dynamic

Device Names for 5250 Emulations in the online documentation for the PowerTerm WebConnect Administration Tool.

- Message Queue** Only visible for Printer emulations.  
Designates to which AS/400 message queue exception messages should be sent. For example: The AS/400 may need to tell the printer to switch to another paper tray.
- Message Library** Only visible for Printer emulations.  
Specifies which library contains the message queue for exception messages.
- Network Name** Only for "Fat Client" (see also Server As Gateway).  
Specifies generic name of the network over which communication to host is being transmitted.

### **MS SNA Server Parameters for 5250**

- System Name** Specifies the name of the system.
- Device Name** When using multiple sessions, each new session can be automatically given a new name, followed by the session number. For example, if the device name is Test, the first session would be Test1, the next session Test2, and so on. For more information see Dynamic Device Names for 5250 Emulations in the online documentation for the PowerTerm WebConnect Administration Tool.
- User Name** Specifies the user name that will be relayed on to the host when attempting to log on.
- Password** Specifies the password that will be relayed on to the

host when attempting to log on.

- Message Queue** Only visible for Printer emulations.  
Designates to which AS/400 message queue exception messages should be sent. For example: The AS/400 may need to tell the printer to switch to another paper tray.
- Message Library** Only visible for Printer emulations.  
Specifies which library contains the message queue for exception messages.
- Network Name** Only for "Fat Client" (see also Server As Gateway).  
Specifies generic name of the network over which communication to host is being transmitted.

### **TELNET Parameters for TANDEM**

- Host Name** Specifies the host computer name or the host's IP address.
- Port Number** Specifies the TELNET port number (default 23).
- Keep Alive Timeout** Periodically sends a NOP (non operation) command to the host when any number greater than zero is entered in this field. For example: Enter 600 (seconds) to "keep alive" every 10 minutes.
- Set Window Size** Tells PowerTerm to report the window size (if asked by the host) in the telnet negotiation procedure.
- Network Name** Only for "Fat Client" (see also Server As Gateway).  
Specifies generic name of the network over which communication to host is being transmitted.

**Type** Specify which SSL protocol to use in the host connection.

**Show Certificate** Displays the Security dialog in which the user can designate which SSL Certificate behavior to implement for your host connection.

**NSVT Parameters**

**Host Name** Specifies the host computer name or the host's IP address.

**Service Name** Specifies the name of the service.

**Network Name** Only for "Fat Client" (see also Server As Gateway). Specifies generic name of the network over which communication to host is being transmitted.

# Index

## A

AboutBox · 10

Application · 5

## B

Browser · 7

## C

CallSupport · 14

ClearScreen · 51

Click · 21

CloseSession · 48

CopyToBitmap · 80

CopyToFile · 79

CurrentURL · 9

## D

DbClick · 22

Display · 65

## E

EnumColorPallette · 94

EnumDisplayAttributes · 91

EnumPrintDevice · 90

EnumPrintScreenConvert · 90

EnumReconnectMode · 89

Exec · 54

## G

GetAppVar · 55

GetColorBackground · 82

GetColorText · 80

GetCursorPos · 77

GetEnvironmentVar · 54

GetPaletteColor · 84

GetPrintDevice · 61

GetPrinterFileName · 59

GetPrinterName · 57

GetPrintScreenConvert · 62

GetRectText · 56

GetScreenText · 55

GetScriptRecording · 88

GetVTButtonAttributes · 84

GetVTEditAttributes · 85

GetWindowSize · 11

## I

InputTrace · 66

IPLogin · 44

IPLoginConnection · 45

IsCommunicationOpen · 49

## K

KeyDown · 16

KeyPress · 17

KeyUp · 16

## L

Left · 8

LockColumns · 65

LoginDialog · 34

LoginDialogConnection · 36

LoginToServer · 31

LoginToServerConnection · 32

## M

MachineLogin · 41

MachineLoginConnection · 42

MapKeyToCommand · 76

MapKeyToDefault · 75

---

MapKeyToNull · 75  
MapKeyToScript · 77  
MapKeyToVTKey · 75  
Message · 66  
MouseDown · 18  
MouseMove · 20  
MouseUp · 19  
O  
OnBlockEnd · 27  
OnCloseSession · 26  
OnConnectToServerBegin · 22  
OnConnectToServerEnd · 23  
OnDisconnectFromServer · 24  
OnOpenSessionBegin · 24  
OnOpenSessionEnd · 25  
OnRecordEnd · 28  
OnSystemEnd · 27  
OpenColorDlg · 12  
OpenKeyMappingDlg · 12  
OpenPowerPadDlg · 12  
OpenPrinterSetupDlg · 13  
OpenSession · 47  
OsLogin · 38  
OsLoginConnection · 39  
P  
PrintFile · 60  
PrintScreen · 60  
R  
RingBell · 57  
RunFTPClient · 14  
RunLPDClient · 13  
RunScriptCommand · 87  
RunScriptFile · 87  
S  
Send · 67  
SendBreak · 68  
SendMail · 15  
SendRawText · 67  
SetColorBackground · 82  
SetColorText · 81  
SetCommonColors · 83  
SetCursorPos · 78  
SetDtr · 51  
SetFocus · 10  
SetFonts · 86  
SetNewCodeData · 77  
SetPowerGui · 79  
SetPrintDevice · 61  
SetPrinterFileName · 59  
SetPrinterName · 58  
SetPrintScreenConvert · 63  
SetRts · 51  
Setup · 4  
SetVTButtonAttributes · 85  
SetVTEditAttributes · 86  
SetWindowSize · 10  
ShowClientToServer · 50  
ShowHistoryScrollBar · 79  
ShowWindow · 11  
Sleep · 68  
StartAutoPrint · 63  
StopAutoPrint · 64  
T  
ToggleAutoPrint · 64

[Top](#) · 8

[U](#)

[UnlockColumns](#) · 65

[UpdateScriptRecording](#) · 88

[UseEmulatorAltKeys](#) · 52

[UseEmulatorCtrlKeys](#) · 52

[UseEmulatorFuncKey](#) · 53

[W](#)

[WaitForBlock](#) · 71

[WaitForCursor](#) · 74

[WaitForRecord](#) · 70

[WaitForSystem](#) · 69

[WaitForText](#) · 72

[WaitForTextOnScreen](#) · 72