

Terminal Viewer Client Functions (API)

This section applies to the COM interface of Entire Screen Builder's Terminal Viewer.

This section assumes that you are familiar with the use of ActiveX controls in applications. You can use ActiveX controls from a wide variety of languages: C++, JavaScript, Visual Basic, Visual Basic for Applications, etc.

The interface consists of a set of properties, methods and events. Properties are needed for the control to know how to make the connection and its visual appearance on screen. The methods are used to handle connection/disconnection and manipulate screen data. The events are used to notify the container about changes in the control's state (e.g. an event is fired when a new screen arrives).

This chapter covers the following topics:

- Methods
- Events

See also: *Common Viewer Functions (API)* and *Overview of Client Control Properties*

Methods

The following methods are available:

- CloseSession
- DisconnectCurrentSession
- GetCursorPosition
- GetScreenSize
- Initialise
- OpenSessionByName
- PerformEditAction
- PrintCurrentScreen
- PrintPreviewCurrentScreen
- PutData
- PutDataMapped
- RunProcedure
- SetCursorPosition
- ShowDialog

CloseSession

This method closes any open host session.

Format

```
boolean CloseSession();
```

Input Parameters

No input parameters.

Return Values

True if successful, false if not.

DisconnectCurrentSession

This method disconnects the control from the server and also closes any host sessions.

Format

```
boolean DisconnectCurrentSession();
```

Input Parameters

No input parameters.

Return Values

True if successful, false if not.

GetCursorPos

This method gets the current cursor position.

Format

```
boolean GetCursorPos(short* Row, short* Column);
```

Input Parameters

`short* Row` The current row position of the cursor.

`short* Column` The current column position of the cursor.

Return Values

True if connected, false if not.

GetScreenSize

This method gets the current screen size. It can be called in response to the ScreenSizeChanged event.

Format

```
boolean GetScreenSize(short* NumRows, short* NumColumns);
```

Input Parameters

`short* NumRows` Current number of rows of the host screen.

`short* NumColumns` Current number of columns of the host screen.

Return Values

True if connected, false if not.

Initialise

This method connects the control to the defined server.

Format

```
boolean Initialise();
```

Input Parameters

No input parameters.

Return Values

True if successful, false if not.

OpenSessionByName

This method opens the named host session which must be defined for the server.

Format

```
boolean OpenSessionByName(BSTR SessionName, BSTR SessionIPAddress, long PortNumber);
```

Input Parameters

BSTR SessionName Defines the host session to be opened.

BSTR SessionIPAddress The IP address for the host session.

long PortNumber The port number for the host session.

If SessionIPAddress and PortNumber are set, they will be used instead of the values defined in the server configuration.

It is not required to set these parameters. If SessionIPAddress is "" and PortNumber is 0, the defined server values will be used.

Return Values

True if successful, false if not.

PerformEditAction

This method automates the edit actions for the control.

Format

```
boolean PerformEditAction(long lEditID);
```

Input Parameters

long lEditID Specify one of the following numbers:

- 0 = Cut
- 1 = Copy
- 2 = Paste
- 3 = Append Copy
- 4 = Clear
- 5 = Select All

Return Values

True if successful, false if not.

PrintCurrentScreen

This method automates printing the current screen.

Format

```
boolean PrintCurrentScreen(short bDirect);
```

Input Parameters

short bDirect	If true, the print dialog will not be shown. If false, the print dialog will be shown before the screen is printed.
---------------	---

Return Values

True if successful, false if not.

PrintPreviewCurrentScreen

This method displays the current screen in print preview mode.

Format

```
boolean PrintPreviewCurrentScreen();
```

Input Parameters

No input parameters.

Return Values

True if successful, false if not.

PutData

This method sends data to the current cursor position. In addition, a function key code can be passed to the server. The keycode will *not* use the mapping defined in the key table for the open session.

The key codes for this method can be found in the *Samples\Definitions* folder of the Entire Screen Builder CD-ROM. The file name is *TerminalViewerKeycodes.h*.

Format

```
boolean PutData(BSTR TextToPut, short KeyCode);
```

Input Parameters

BSTR TextToPut	Text to be copied to the current screen position.
short KeyCode	The function key code to be passed to the server.

Return Values

True if successful, false if not.

PutDataMapped

This method sends data to the current cursor position. In addition, a function key code can be passed to the server. The keycode will use the mapping defined in the key table for the open session.

The key codes for this method can be found in the *Samples\Definitions* folder of the Entire Screen Builder CD-ROM. The file name is *TerminalViewerKeycodes.h*.

Format

```
boolean PutDataMapped(BSTR TextToPut, short MappedKeyCode);
```

Input Parameters

BSTR TextToPut	Text to be copied to the current screen position.
short MappedKeyCode	The function key code to be passed to the server.

Return Values

True if successful, false if not.

RunProcedure

This method attempts to start the named script on the server.

Format

```
boolean RunProcedure(BSTR sProcedureName);
```

Input Parameters

BSTR sProcedureName	Must contain the name of the script on the server.
---------------------	--

Return Values

True if successful, false if not.

SetCursorPos

This method changes the current cursor position.

Format

```
boolean SetCursorPos(short Row, short Column);
```

Input Parameters

short Row Row to which the cursor is to be set.

short Column Column to which the cursor is to be set.

Return Values

True if connected, false if not.

ShowDialog

This method is used to automatically show the control's configuration dialogs.

Format

```
boolean ShowDialog(long lDialogID);
```

Input Parameters

long lDialogID Specify one of the following numbers:

- 0 = Open session dialog
- 1 = Task list dialog
- 2 = Color dialog
- 3 = Font dialog

Return Values

True if successful, false if not.

Events

Events are used to notify the container of changes in the state of the control. The following events are available:

Event	Description
void Connected()	This event is fired when the control is connected to the server.
void Disconnected()	This event is fired when the control is disconnected from the server.
void ScreenSizeChanged()	This event is fired when the screen size in the open session has changed (e.g. when the connected session changes from mode 3 to mode 4).
void SessionClosed()	This event is fired when an active host session has been closed.
void SessionOpened()	This event is fired when an active host session has been opened.