PASSPORT
Automation Interface

The PASSPORT Automation Interface Provides Integration of Terminal Emulation with Desktop Applications

The Automation Interface was added to PASSPORT PC TO HOST in version 2004 SP-1 and is a special interface that can be used to integrate terminal emulation inside a master desktop application. The master desktop application would typically be used to manage several host sessions concurrent with other desktop applications.

Using the host application screen display as an API, developers can quickly navigate through TN3270, TN5250, VT100/VT220 and SCO ANSI applications to locate specific screens, fields and text, and cut and paste data to and from the host session.

A lightweight screen scraping API is provided, as well as methods to execute any PASSPORT menu command, action, macro or external program(s). This interface method provides seamless integration with the PASSPORT terminal emulator.

The PASSPORT Automation Interface provides the ability for PASSPORT PC TO HOST to run as an ActiveX Document inside of the Internet Explorer control. The IE control is hidden and then placed within an application. C++ or any programming language that can host the IE control can use this interface. The automation interface can only be used with PASSPORT PC TO HOST and cannot be used with PASSPORT WEB TO HOST.

When to Use the Automation Interface API

The Automation Interface is recommended when a terminal emulator needs to be integrated into a 'master' desktop application which controls host access and execution of other applications.

GetRows

DESCRIPTION

Returns the number of rows in the presentation space.

SYNTAX

ret = GetRows ( )

PARAMETERS

ret Integer, output
The number of rows in current screen (either 24, 32, 43 or 27)

REMARKS

This function returns the number of rows in the PASSPORT terminal emulator screen. There are four screen sizes supported: model 2, 3, 4 and 5. The screen size determines the number of columns and rows are on the display.

3270 SCREEN SIZES

1920 = Model 2 (80 columns x 24 rows)
2560 = Model 3 (80 columns x 32 rows)
3440 = Model 4 (80 columns x 43 rows)
3564 = Model 5 (132 columns x 27 rows)

5250 SCREEN SIZES

1920 = Model 2 (80 columns x 24 rows)
3564 = Model 5 (132 columns x 27 rows)

GetCols

DESCRIPTION

Returns the number of columns in the presentation space.

SYNTAX

ret = GetCols ( )

PARAMETERS

ret Integer, output
The number of columns in current screen (either 80 or 132)

REMARKS

This function returns the number of columns in the PASSPORT terminal emulator screen. See remarks for GetRows for more detail.

GetCursorRow

DESCRIPTION

Returns the current row position of the cursor

SYNTAX

ret = GetCursorRow ( )

PARAMETERS

ret Integer, output
The row position of the cursor.

REMARKS

The row position ranges in value from 1 to 24 for a model 2 screen size, 1 to 32 for a model 3 screen size, 1 to 43 for a model 4 screen size, and 1 to 27 for a model 5 screen size.

GetCursorCol

DESCRIPTION

Returns the current column position of the cursor

SYNTAX

ret = GetCursorCol( )

PARAMETERS

ret Integer, output
The column position of the cursor.

REMARKS

The column position ranges in value from 1 to 80 for model 2, 3 and 4 screen sizes. The column position ranges in value from 1 to 132 for the wide column model 5 screen size. For VT screen sizes the column position ranges 1 to 80 or 1 to 132.

SetCursor

DESCRIPTION

Position the cursor at the new position

SYNTAX

SetCursor (row, column)

PARAMETERS

row Integer, input
The new row position of the cursor

column Integer, input
The new column position of the cursor.

REMARKS

This function moves the cursor to a new position specified by a row and column position. Row 1 column 1 indicates the home position in the upper left corner of the screen. Care must be taken not to position the cursor off the screen.

GetString

DESCRIPTION

Returns a string from the PASSPORT terminal emulation screen

SYNTAX

ret = GetString (row, column, length)

PARAMETERS

ret String, output
The string to contain the data read from the PASSPORT screen.

row Integer, input
The starting row position in the PASSPORT screen.

column Integer, input
The starting column position in the PASSPORT screen.

length Integer, input
The maximum number of characters to read.

REMARKS

This function reads a string from the PASSPORT terminal emulator screen at the specified position. Row 1 column 1 indicates the home position in the upper left corner of the screen.

SendString

DESCRIPTION

Copies a string into the PASSPORT terminal emulation screen, to later be sent back to the host mainframe.

SYNTAX

SendString (row, column, string)

PARAMETERS

row Integer, Input
The starting row position in the PASSPORT screen.

column Integer, Input
The starting column position in the PASSPORT screen.

string String, Input
The string to copy into the PASSPORT terminal emulation screen.

REMARKS

This function copies a string into the PASSPORT terminal emulator screen at the specified row and column position. Row 1 column 1 indicates the home position in the upper left corner of the screen.

Strings should only be copied into unprotected fields. When copied into an unprotected field, the MDT (Modified Data Tag) is set in that field so that the data is transmitted to the host whenever the ENTER key or other inbound AID key is sent.

SendKeys

DESCRIPTION

Sends ANSI characters to the host mainframe

SYNTAX

SendKeys (string)

PARAMETERS

string String, input
This string should include only ANSI characters

REMARKS

The string of keys can be any ANSI characters. If you want send AID key, you should use SendAID method.

SendAID

DESCRIPTION

Sends host key to the presentation space.

SYNTAX

SendAID (aidKey)

PARAMETERS

aidKey Integer, input
Integer value representing the host key

REMARKS

The key can be either local key(like TAB) or AID key which will transmit data back to the host. See Pass.h for list of local and AID keys.

GetOIA

DESCRIPTION

Returns the text in the Operator Information Area (OIA)

SYNTAX

ret = GetOIA ( )

PARAMETERS

ret String, output
The text returned from the OIA.

REMARKS

Returns the text in OIA line of presentation space.

ExecMenuCmd

DESCRIPTION

Sends menu command to PASSPORT program.

SYNTAX

ret = ExecMenuCmd (cmdID)

PARAMETERS

ret Boolean, output
True if it is a valid menu command, otherwise False

cmdID Integer, input
Integer value representing PASSPORT menu command

REMARKS

Menu command is always in the format of 0x02xx. See Pass.h for list of menu commands.

RunMacro

DESCRIPTION

Run the specified macro on PASSPORT application

Zephyr Development Corporation specializes in advanced IP host access solutions for Microsoft Windows desktops and servers, including Windows 2003, XP and 2000. Using TN3270, TN5250, VT100, VT220, SCO ANSI and FTP standards, we offer PC and web terminal emulation, toolkits and programmatic API modules, allowing our customers to connect and integrate proven legacy applications with other PC, server and .NET or ActiveX applications.

THE AMERICAS

8 E GREENWAY PLAZA, SUITE 1414
HOUSTON, TX 77046 USA

800-966-3270
TEL: 1-713-623-0089
FAX: 1-713-623-0091
info@zephyrcorp.com

UK AND EMEAI

71 HIGH STREET
HARROLD, BEDFORDSHIRE MK43 7BJ UK

TEL: 44 (0) 1234 721755
FAX: 44 (0) 1234 721672

zephyr@integranet.co.uk

ZEPHYR

SYNTAX

RunMacro (string)

PARAMETERS

string String, input
This string is the full path of the macro

REMARKS

You should always specify the full path for the macro.

ExecCmd

DESCRIPTION

Execute command same as Keyboard, Keypad, Toolbar etc. in PASSPORT application

SYNTAX

ExecCmd (type, string)

PARAMETERS

type Integer, Input
The type of command can be executed in PASSPORT

string String, Input
The string needed for executing the command

REMARKS

The type of command which can be executed in PASSPORT include

type	Type value	string
Text Character	Character value (byte)	Not needed
Text String	0x400	Text string to send to screen
Host Key	0x1xx	Not needed
Dead Key	0x6xx	Not needed
Macro	0x500	Full path of the macro
Menu Command	0x2xx	Not needed
Miscellaneous	0x7xx	May be needed depending on the action

See Pass.h for list of Host keys, Dead keys, Menu commands and miscellaneous actions.