

Host Communication Methods

The following host communication methods are available:

- CloseSession
 - GetDefaultSession
 - GetDisplay
 - GetScreenText
 - OpenSession
 - OpenUpdatedSession
 - SendKey
 - SetDataTransfer
 - SetDisplay
 - SetSessionParameter
 - SetUnixLogonCredentials
 - SetXMLEncoding
 - SetXMLStyleSheet
 - TypeString
 - UseDefinedSession
-

CloseSession

Close the current session.

Syntax

```
CloseSession()
```

Parameters

None.

Description

The `CloseSession` method closes the current host session.

If you want to close a session with the `CloseSession` method and after this open another session with `OpenSession()` or `OpenUpdatedSession()`, you should make sure that there is a delay of at least 2 seconds between these two statements. This delay is required because `CloseSession()` does not synchronize. A delay can be defined, for example, with the statement `NSW.Pause (2000)`.

Example

```
NSW.CloseSession();
```

Return Values

None.

GetDefaultSession

Get the default session.

Syntax

```
GetDefaultSession()
```

Parameters

None.

Description

The `GetDefaultSession` method returns the name of the defined default session for the current user. This value can then be used to open a session using the methods `UseDefinedSession` or `OpenSession`.

Example

```
szSession = NSW.GetDefaultSession();
```

Return Values

Value of the variable. If the variable was not found, an empty string is returned and `errno` is set to `PROC_USER_HAS_NO_DEFINED_SESSION`.

GetDisplay

Return the current state of the display flag.

Syntax

```
GetDisplay( )
```

Parameters

None.

Description

The GetDisplay method returns the current state of the display flag which has been set with the SetDisplay method.

Example

```
iDispValue = NSW.GetDisplay();
```

Return Values

Either DISPLAY_ON or DISPLAY_OFF.

GetScreenText

Get text from the terminal emulation screen.

Syntax

```
GetScreenText(StartRowPos, StartColPos, Length)
```

Parameters

StartRowPos Values from 1 through NSW.maxRows.

StartColPos Values from 1 through NSW.maxColumns.

Length Values from 1 through NSW.maxRows * NSW.maxColumns.

Description

The GetScreenText method gets the text from the current terminal emulation screen starting in *StartRowPos* and *StartColPos*. The length of this text is determined by the *Length* parameter.

Example

Get the contents of the complete terminal emulation screen:

```
strValue = NSW.GetScreenText(1,1,NSW.maxRows*NSW.maxColumns);
```

Return Values

A string.

OpenSession

Open a session.

Syntax

```
OpenSession(Name[, Keyword, Value]...)
```

Parameters

Name A valid session name.

Keyword The parameter to be changed.

Value The new value for the parameter.

Description

The OpenSession method opens the host session with the specified session name.

You can specify more than one value pair consisting of *Keyword* and *Value*.

Optionally the session parameters can be changed. See the description of the SetSessionParameter method for a list of all valid parameters.

Examples

```
Result = NSW.OpenSession("Darmstadt Host");
Result = NSW.OpenSession("Darmstadt Host", HOST_TCPIP_PORT, 23);
Result = NSW.OpenSession("Darmstadt Host", HOST_TCPIP_PORT, 23, HOST_TN3270E, true);
```

Return Values

True if the method was successful. False if there was an error.

OpenUpdatedSession

Open the changed session.

Syntax

```
OpenUpdatedSession()
```

Parameters

None.

Description

The `OpenUpdatedSession` method opens the host session which was defined with a previous call to the method `UseDefinedSession` and where the parameters have been changed with the method `SetSessionParameter`.

This method only makes sense when being used together with the methods `UseDefinedSession` and `SessionParameter`.

Example

```
NSW.OpenUpdatedSession();
```

Return Values

True if the method was successful. False if there was an error.

If `UseDefinedSession` has not been called previously, this method returns false and `errno` is set to `NSW_NO_SESSION_SPECIFIED`.

SendKey

Send a terminal emulation key to the host session.

Syntax

```
SendKey(Row, Column, Keycode)
```

Parameters

<i>Row</i>	The number of the row to which the cursor is to be moved. Or -1 for the current position of the terminal emulation cursor.
<i>Column</i>	The number of the column to which the cursor is to be moved. Or -1 for the current position of the terminal emulation cursor.
<i>Keycode</i>	A terminal emulation key. See <i>Terminal Emulation Keys</i> in Entire Screen Builder's <i>System Management Hub</i> documentation.

Description

The SendKey method sends the specified terminal emulation key to the current host session. If both *Row* and *Column* are positive values, the cursor is moved to the specified position first. The SendKey method then waits for a response from the host before processing continues with the next instruction.

To send PF keys in AS/400 style, the session has to be defined as an AS/400 session. See *Communication Properties for Telnet TN3270* in Entire Screen Builder's *System Management Hub* documentation.

Examples

```
Result = NSW.SendKey(-1, -1, PF3);
Result = NSW.SendKey(-1, -1, AS_PF3);
```

Return Values

True or false. In case of an error, *errno* is set to 1.

SetDataTransfer

Set the PC file name for Natural data transfer.

Syntax

```
SetDataTransfer(SetCommand)
```

Parameters

SetCommand A valid SET command for data transfer:

```
SET PCFILE x DOWN DATA filename.ext
SET PCFILE x UP DATA filename.ext
SET PCFILE y DOWN REPORT filename.ext
```

Description

With the SetDataTransfer method, you can set the file name for subsequent uploads or downloads. *x* is a valid work file number. *y* is a valid printer file number. *filename.ext* is a file on the client workstation.

To reset the number for a work file or printer file, use the SET command without specifying a file name.

See also: *Specifying a File Name Using the SET Command* in the *Data Transfer* documentation.

Example

```
Result = NSW.SetDataTransfer("SET PCFILE 7 DOWN DATA C:\\TEMP\\TEST.NCD");
```

Return Values

True if the method was successful. False if there was an error.

SetDisplay

Stop the screen being updated in the client.

Syntax

```
SetDisplay(Flag)
```

Parameters

Flag Either DISPLAY_OFF or DISPLAY_ON.

Description

DISPLAY_OFF stops the screen being updated in the client until the next call to DISPLAY_ON.

If the script ends in DISPLAY_OFF mode, the screen is automatically refreshed.

Example

```
NSW.SetDisplay(DISPLAY_OFF);
```

Return Values

True if the method was successful. False if there was an error.

SetSessionParameter

Change the parameters of the selected session temporarily.

Syntax

```
SetSessionParameter(Keyword, Value[, Keyword, Value]...)
```

Parameters

Keyword The parameter to be changed. See the table below for a list of valid keywords.

Value The new value for the parameter.

Description

With the `SetSessionParameter` method, you can change the parameters of the session that has previously been selected with the `UseDefinedSession` method. The changed session can then be opened by calling the method `OpenUpdatedSession`.

You can specify more than one value pair consisting of *Keyword* and *Value*.

Important:

The parameters changed with `SetSessionParameter` are only changed inside the JavaScript code. Changing session parameters here has no effect on the "real" session as defined with the System Management Hub.

You can redefine the session temporarily, without having to change the settings in the System Management Hub and without having to restart the server. For example, you can prompt for the input of a user name and a password and then use the obtained values for the session by assigning them to `HOST_USER_ID` and `HOST_PASSWORD`.

The following table lists all valid keywords and indicates the appropriate type(s) of host session.

Keyword	Type	Description	Telnet TN3270	Telnet VT	BS2000	Natural UNIX
HOST_TCPIP_PORT	Integer	TCP/IP port.	X	X	X	X
HOST_TCPIP_ADDR	String	TCP/IP address.	X	X	X	X
HOST_TERMINAL_TYPE	String	Terminal type.	X Valid values: IBM-3278-2 IBM-3278-4 IBM-3278-2-E IBM-3278-4-E IBM-3279-2 IBM-3279-4 IBM-3279-2-E IBM-3279-4-E IBM-3278-3 IBM-3278-5 IBM-3278-3-E IBM-3278-5-E IBM-3279-3 IBM-3279-5 IBM-3279-3-E IBM-3279-5-E	X Valid values: VT100 VT220 VT320		
HOST_IGNORE_EABS	Boolean	Ignore extended attribute bytes in data stream.	X			
HOST_TN3270E	Boolean	Open the session as a TN3270E session.	X			
HOST_DEVICE_NAME	String	Use device name for a TN3270E session.	X			
HOST_EOS_DELAY	Integer	End-of-screen delay.	X	X	X	
HOST_USER_ID	String	User name.				X
HOST_PASSWORD	String	Password (for user).				X
HOST_SHOW_NILS	Boolean	Display NIL characters (dots) in BS2000.			X	
HOST_BS2000_APPLIC	String	BS2000 application.			X	
HOST_STATION_NAME	String	BS2000 station name.			X	
HOST_COMPRESSED	Boolean	Compressed session.				X
HOST_LOGON_CREDENTIALS	Boolean	Logon credentials.				X
HOST_VT_LINE	Integer	Line size: 80 or 132.		X		

Keyword	Type	Description	Telnet TN3270	Telnet VT	BS2000	Natural UNIX
HOST_VT_CR_OPTION	Constant	Return key send option		X Valid values: VT_CR_OPTION_PLAIN_CR (Plain CR) VT_CR_OPTION_CR_LF (CR + LF) VT_CR_OPTION_CR_NULL (CR + NULL)		

Examples

```
Result = NSW.SetSessionParameter(HOST_TERMINAL_TYPE, "IBM-3279-2-E");
Result = NSW.SetSessionParameter(HOST_TCPIP_PORT, 23, HOST_TN3270E, true);
```

Return Values

True if the method was successful. False if there was an error.

If `UseDefinedSession` has not been called previously, this method returns false and `errno` is set to `NSW_NO_SESSION_SPECIFIED`.

SetUnixLogonCredentials

Terminal Viewer only.

Force a UNIX logon before a session can be opened.

Syntax

```
SetUnixLogonCredentials()
```

Parameters

None.

Description

Use this method before an OpenSession call.

If a JavaScript method is opening a Natural UNIX session that requires user logon credentials, SetUnixLogonCredentials must be called first. Otherwise, the dialog prompting for user name and password is not shown.

Example

```
bUserClickedOK = NSW.SetUnixLogonCredentials();
```

Return Values

True if the user has chosen the **OK** button. False if the user has not chosen the **OK** button or if the function is not supported.

SetXMLEncoding

Set encoding for data transfer with HTML and XML.

Syntax

```
NSW.SetXMLEncoding (Encoding)
```

Parameters

Encoding The XML/HTML encoding.

Description

The SetXMLEncoding method defines the encoding that is to be used for data transfer with HTML and XML. When a connection has been established with a viewer, this method temporarily overwrites the setting made in the session definition. When the viewer is disconnected, the encoding from the session definition is used again. See *General Properties* in the *Host Sessions* section of Entire Screen Builder's *System Management Hub* documentation.

See also the following sections in the *Data Transfer* documentation:

- *Downloading Data to HTML*
- *Downloading Data to XML*
- *Uploading XML Files*

The following example shows the header for an XML file that results from executing SetXMLEncoding ("utf-8"):

```
<?xml version="1.0" encoding="utf-8"?>
```

To cancel the usage of the encoding, you have to call SetXMLEncoding with an empty string parameter. In this case, the above XML header is written with the following default encoding:

```
<?xml version="1.0" encoding="windows-1252"?>
```

Examples

```
// Add encoding definition  
NSW.SetXMLEncoding ("utf-8")  
  
// Reset to default encoding  
NSW.SetXMLEncoding ("")
```

Return Values

None.

SetXMLStyleSheet

Define style sheet for XML download.

Syntax

```
NSW.SetXMLStyleSheet (Type, HRef)
```

Parameters

Type The type of XML style sheet.

HRef The location of the style sheet.

Description

The SetXMLStyleSheet method defines the style sheet that is to be used when downloading data to an XML file (see *Downloading Data to XML* in the *Data Transfer* documentation). After executing this method, all downloaded XML files contain a style sheet definition like the following:

```
<?xml:stylesheet type="text/xsl" href="http://pcnha/xml/employ2.xsl"?>
```

To cancel the usage of the style sheet, you have to call SetXMLStyleSheet with empty string parameters. In this case, the style sheet definition is no longer written to the downloaded XML files.

Note:

It is also possible to specify a style sheet using the SET command. The last command that was issued (either SetXMLStyleSheet or SET) always determines the style sheet to be used.

Examples

```
// Insert style sheet definition
NSW.SetXMLStyleSheet ("text/xsl", "http://pcnha/xml/employ2.xsl")

// Cancel usage of style sheet
NSW.SetXMLStyleSheet ("", "")
```

Return Values

None.

TypeString

Send a string or a string plus a terminal emulation function key to the current host session.

Syntax

```
TypeString( Row, Column, String, [Keycode] )
```

Parameters

<i>Row</i>	A number value. Valid values are 1 through NSW.maxRows, or -1.
<i>Column</i>	A number value. Valid values are 1 though NSW.maxColumns, or -1.
<i>String</i>	A string value.
<i>Keycode</i>	A terminal emulation key. See <i>Terminal Emulation Keys</i> in Entire Screen Builder's <i>System Management Hub</i> documentation.

Description

The string is typed into the host session at the specified cursor position.

If both values for *Row* and *Column* are -1, the string is typed at the current cursor position and the cursor is moved. For sessions of type Natural UNIX, this does not work in all situations. It is recommended that you always use the exact cursor position when working with this type of session.

If a valid cursor position is given in *Row* and *Column*, the string is typed at this position and the cursor is moved. Note that the cursor must be in an unprotected (input) field to allow typing in.

For sessions of type Natural UNIX, this call only makes sense, if *Keycode* is provided. Otherwise, "CR" is sent.

Examples

```
NSW.TypeString(-1, -1, "MyUserID");
NSW.TypeString(-1, -1, "MyUserID", CR);
NSW.TypeString(2, 11, "abcd");
```

Return Values

True or false. In case of an error, errno is set.

UseDefinedSession

Select a session.

Syntax

```
UseDefinedSession(Name)
```

Parameters

Name A valid session name.

Description

The UseDefinedSession method selects the session to be used with the methods SetSessionParameter and OpenUpdatedSession.

This call only makes sense when called together with SetSessionParameter and OpenUpdatedSession.

Example

```
Result = NSW.UseDefinedSession("Darmstadt Host");
```

Return Values

True if the method was successful. False if there was an error.