



z/Scope Terminal Emulator

Administrator's and User's guide

Table of Contents

Overview	11
What's New	12
Getting to know z/Scope 6.2	12
1 The User Interface	13
Start Page	13
Tab Selection Bar	14
Tiled View	15
Telnet Workspace	16
Main Menu	17
Main Toolbar	22
Emulation Display	26
Printer Session Panel	28
Print Job Queue	29
Print Preview Dialog	30
Printer Parameters Dialog	32
Status Bar	33
FTP Workspace	34
Main Menu	34
Main Toolbar	38
My Computer Panel	40
FTP Server Panel	41
File Transfer Queue	43
Queue Item Edit	45
Log Window	46
Status Bar	46
Web Browser Workspace	47
Main Menu	47
Main Toolbar	53
Browser Panel	54
Status Bar	54
2 Key Features	55
Auto-fills	55
Creating an Auto-Fill	55
Using an Auto-Fill	58
Auto-suggestion	59
Using Auto-suggestion	60
Deleting Auto-suggestions	60
Favorites	61
Building Your Favorites List	62
Organizing Your Favorites List	65
File Transfer	67
Using the File Transfer Manager	67
FTP	70
IND\$FILE	76
KERMIT	82
X-MODEM	84
Y-MODEM	87
Z-MODEM	91

Using the FTP Workspace.....	94
HotSpots	97
Creating a HotSpot.....	97
Using HotSpots.....	101
Keyboard Support	102
Keypads	103
Using Keypads.....	103
Macros	104
Creating a Macro.....	104
Using Macros	105
MS Office Integration	106
Exporting to Microsoft® Word.....	106
Exporting to Microsoft® Outlook.....	106
Navigators	107
Creating a Navigator.....	108
Running a Navigator.....	112
OHIO	114
HLLAPI	115
Screen History	115
Screen Styles	117
Using Screen Styles.....	117
Using z/Scope Skins	118
Security and Encryption	119
Using SSL.....	120
3 Setting up a Connection.....	121
Setting up a UNIX/VT/SSH Connection	123
Connection with SSH.....	125
Connection with HTTP Gateway.....	128
Setting up an IBM Mainframe or AS/400 Connection	130
Connection with SSH.....	133
Connection with HTTP Gateway.....	136
Setting up an FTP Connection	138
Settings	142
1 Connections.....	144
Unix/VT Connections	146
General.....	147
Host	148
Backup	149
Display	151
Printer	152
Socks	153
SSL	155
SSH	156
Gateway.....	157
Preferences	157
Options	159
HotSpots.....	160
Mainframe / AS400 Connections	161
General.....	162
Host	163
Backup	164
Display	166

Printer	167
AS/400	167
Mainframe	169
Socks	170
SSL	171
Gateway.....	172
Preferences.....	172
Char Table.....	174
HotSpots.....	175
Keypads.....	175
FTP	176
General.....	177
Host	177
Options.....	179
SSL	180
Remote Desktop	182
General.....	182
Host	183
Authentication.....	183
Display.....	184
Programs.....	185
Local Resources.....	186
Experience.....	187
VNC	189
General.....	189
Host	190
Authentication.....	191
Options.....	191
Deleting a Connection	192
2 Screen Styles.....	193
Creating/Editing Screen Styles	194
General.....	195
Cursor	196
Font	197
Field Options.....	199
Color Mapping.....	199
5250 Attributes.....	200
VT Defaults.....	201
Deleting Screen Styles	202
3 Hotspots.....	203
Creating/Editing a Hotspot	204
General.....	205
Rules	206
Style	210
Deleting a Hotspot	211
4 Keypads.....	211
Creating/Editing a Keypad	212
Deleting a Keypad	215
5 Macros.....	215
Editing Macros	217
Debugging Macros	219
Deleting Macros	221

6 Navigators.....	221
Editing a Navigator	222
General.....	224
Screen Fields	225
Visualization.....	226
Deleting a Navigator	227
7 Auto-Fills.....	227
Editing an Auto-Fill	228
General.....	230
Screen Fields	231
Visualization.....	232
Deleting an Auto-fill	233
8 Keyboard.....	233
Base	234
Host	235
Edition	237
Char	238
Custom	240
Keyboards	242
z/Scope Keyboard Map.....	242
IBM Keyboard Map.....	244
EXTRA Keyboard Map.....	246
IRMA Keyboard Map.....	249
RUMBA Keyboard Map.....	252
9 Environment.....	255
Edit	256
Display	258
Misc	259
Plugins	261
File Transfer	262
Start Page	262
Associations	264
Debug	265
Config Setup.....	266
Programming Reference	266
1 Scripting Objects.....	267
Display Object	267
Properties.....	267
Cols	268
Row s	268
CursorPos	268
EditFields	269
HostFields	269
HTML	270
ScreenId	270
ScreenRow	271
Text	271
Methods.....	272
Connect	272
Disconnect	273
Press	273

PressAndWait	273
SendAid	274
SendKeys	275
Wait	276
WaitFor	276
WaitForConnect.....	277
WaitForDisconnect.....	277
WaitForScreen	278
WaitForNew Screen.....	278
WaitForUnlock	279
WaitForNew Unlock.....	279
2 HLLAPI Reference.....	280
EHELLAPI Functions	280
Cancel File Transfer (92).....	281
Change PS Window Name (106).....	282
Change Switch List LT Name (105).....	283
Connect Presentation Space (1).....	284
Connect Window Services (101).....	286
Convert Position or Convert RowCol (99).....	287
Copy Field to String (34).....	289
Copy OIA (13).....	291
Copy Presentation Space (5).....	292
Copy Presentation Space to String (8).....	293
Copy String to Field (33).....	295
Copy String to Presentation Space (15).....	296
Disconnect Presentation Space (2).....	298
Disconnect Window Service (102).....	299
Find Field Length (32).....	300
Find Field Position (31).....	301
Get Key (51).....	303
Lock Presentation Space API (60).....	305
Lock Window Services API (61).....	307
Pause (18).....	308
Post Intercept Status (52).....	310
Query Additional Field Attribute (45).....	311
Query Close Intercept (42).....	312
Query Communication Event (81).....	312
Query Cursor Location (7).....	313
Query Field Attribute (14).....	314
Query Host Update (24).....	316
Query Session Status (22).....	317
Query Sessions (10).....	318
Query System (20).....	320
Query Window Coordinates (103).....	321
Receive File (91).....	322
Release (12).....	325
Reserve (11).....	325
Reset System (21).....	326
Search Field (30).....	327
Search Presentation Space (6).....	329
Send File (90).....	331
Send Key (3).....	333
Set Cursor (40).....	335
Set Session Parameters (9).....	336

Start Close Intercept (41).....	342
Start Communication Notification (80).....	344
Start Host Notification (23).....	346
Start Keystroke Intercept (50).....	349
Start Playing Macro (110).....	351
Stop Close Intercept (43).....	351
Stop Communication Notification (82).....	352
Stop Host Notification (25).....	353
Stop Keystroke Intercept (53).....	354
Wait (4)	355
Window Status (104).....	355
WinHLLAPI Extension Functions	358
Asynchronous Functions.....	358
WinHLLAPIAsync.....	358
Receive File (91).....	359
Start Close Intercept (41).....	360
Send File (90)	361
Start Host Notification (23).....	363
Start Keystroke Intercept (50).....	364
Wait (4)	364
WinHLLAPICancelAsyncRequest.....	365
Initialization and Termination Functions.....	366
WinHLLAPI Startup.....	366
WinHLLAPI Cleanup.....	367
Blocking Routines	367
WinHLLAPIsBlocking.....	368
WinHLLAPISetBlockingHook.....	368
WinHLLAPIUnhookBlockingHook.....	369
WinHLLAPICancelBlockingCall.....	370
3 OHIO Reference.....	370
Examples	370
Visual Basic.....	371
Html	376
Delphi	378
OHIO Containment Hierarchy	384
OHIO Inheritance Hierarchy	385
Programming Reference	385
Class Definition.....	385
OhioField	385
Methods	385
GetData	386
Properties	386
Attribute	386
EndPos	386
Hidden	387
HighIntensity	387
Length	387
Modified	388
Numeric	388
PenSelectable	388
ProtectedField	389
StartPos	389
String/Text	389

OhioFields	390
Methods	390
FindByPosition	390
FindByString	391
Item	391
Refresh	391
Properties	392
Columns	392
Count	392
Rows	393
OhioManager	393
Methods	393
CloseSession	393
OpenSession	394
Properties	395
Sessions	395
OhioOIA	395
Properties	395
AlphaNumeric	395
CommCheckCode	396
InputInhibited	396
MachineCheckCode	396
Numeric	397
Owner	397
ProgCheckCode	397
OhioPosition	398
Methods	398
Refresh	398
Properties	398
Col	398
Pos	399
Row	399
OhioScreen	399
Events	400
OnConnect	400
OnConnectionFail	400
OnDisconnect	400
OnScreenChange	400
OnSendAid	400
OnSessionState	400
Methods	401
FindString	401
GetData	401
Refresh	402
SendAid	402
SendKeys	403
SetString	403
Wait	404
WaitFor	404
WaitForNew Screen	405
WaitForNew Unlock	405
WaitForScreen	405
WaitForUnlock	406
Properties	406

Columns	406
Cursor	407
Fields	407
OIA	407
Row s	408
String/Text	408
OhioSession	408
Events	409
OnConnect	409
OnConnectionFail.....	409
OnDisconnect	409
OnScreenChange.....	409
OnSendAid	409
OnSessionState.....	409
Methods	409
Connect	410
Disconnect	410
WaitForConnect.....	410
WaitForDisconnect.....	411
Properties	411
ConfigurationResource.....	411
Connected	411
Manager	412
Screen	412
SessionName	412
SessionState	413
SessionType	413
Trace	413
Syntax Conventions.....	413
OhioSessions	414
Methods	414
AddSession	414
CloseSession	415
Properties	415
Count	415
Item	415
Manager	416
Constants.....	416
3270 Key Values	416
5250 Key Values	417
Ohio Constants	418
OHIO Base Class.....	422
Methods	422
CreateOhioPosition.....	422
Properties	422
OhioVersion	422
VendorName	423
VendorObject	423
VendorProductVersion.....	423
Syntax Conventions.....	424
4 DDE Reference.....	424
DDE Keystroke Table	425
Appendix A - Character Conversion Tables	426

1	Internal Conversion Tables.....	427
2	Modifying the Character Table.....	429
3	Using an external Character Table.....	430
	Appendix B - Regular Expressions	431
	Appendix C - Useful setup installation command-line switches	433
	Appendix D - Web Server Configuration	436
	Appendix E - Plugins	438
	Purchasing z/Scope	439
1	Licensing Information.....	439
2	Registering z/Scope Trial Version.....	440
3	How to Place an Order.....	441
	Obtaining Technical Support	442

1 Overview



[z/Scope 6.2](#) is the newest version of the z/Scope terminal emulation product family for accessing IBM Mainframes, IBM AS/400 and Unix servers. It can be run under Windows Vista, Windows 7, Windows ME, Windows NT/2000 and Windows XP platforms, supporting TN3270E, TN5250E and VT100/220/320/420 terminal emulation protocols.

A fully integrated FTP client is also provided, with full awareness of AS/400 objects and z/OS datasets.

You can also browse web pages without leaving the z/Scope environment thanks to its integrated web browsing interface.

Remote Printing Services are can be included to make host print-jobs available to local and network printers by emulating IBM-3287-1 and IBM-3812-1 printers.

z/Scope offers all the necessary tools to access IBM Mainframes, As/400 and Unix.

Thanks to our product family variety there is an appropriate solution to suite all needs, so you can do your job easier and faster.

Disclaimer: The product names and logos used in this help are for identification purposes only. All trademarks, registered trademarks and logos are the property of their respective owners.

Copyright © 2002-2014, Cybele Software, Inc. All rights reserved.

2 What's New

What's new in version 6.2

- **New!** Remote Desktop connectivity
- **New!** VNC connectivity
- **New!** DDE Support
- **New!** Multiple Keyboard Profiles
- **New!** Web Update

What's new since version 6.1

- File Transfer Manager.
- Improved FTP client.

What's new since version 6.0

- Extended terminal emulation capabilities, fully VT100/220/320/420 compatible.
- SSH 1 & 2.
- Start page, with RSS aggregator, features links for configured connections.
- Configure this multi-personality application through our innovational plugin system to adjust it to your daily needs.
- Up to 4 emulations visible in the same screen using a tile system.
- Create and organize your own Favorite list for web sites, feeds and connections.
- Embellish your screens using our new z/Scope Font.
- The user interface has been re-designed from ground into four different but at the same time fully integrated workspaces. In this way you can instantly switch between all your open Telnet or FTP sessions and web pages using tabs.

3 Getting to know z/Scope 6.2

This section is intended to provide first-time users an initial approach to the basic functionality of z/Scope Terminal Emulator.

Browse though the following sections to start discovering all that z/Scope can do for you!

- [The User Interface](#)
- [Key Features](#)
- [Setting up a connection](#)
- [Configuration Settings](#)

3.1 The User Interface

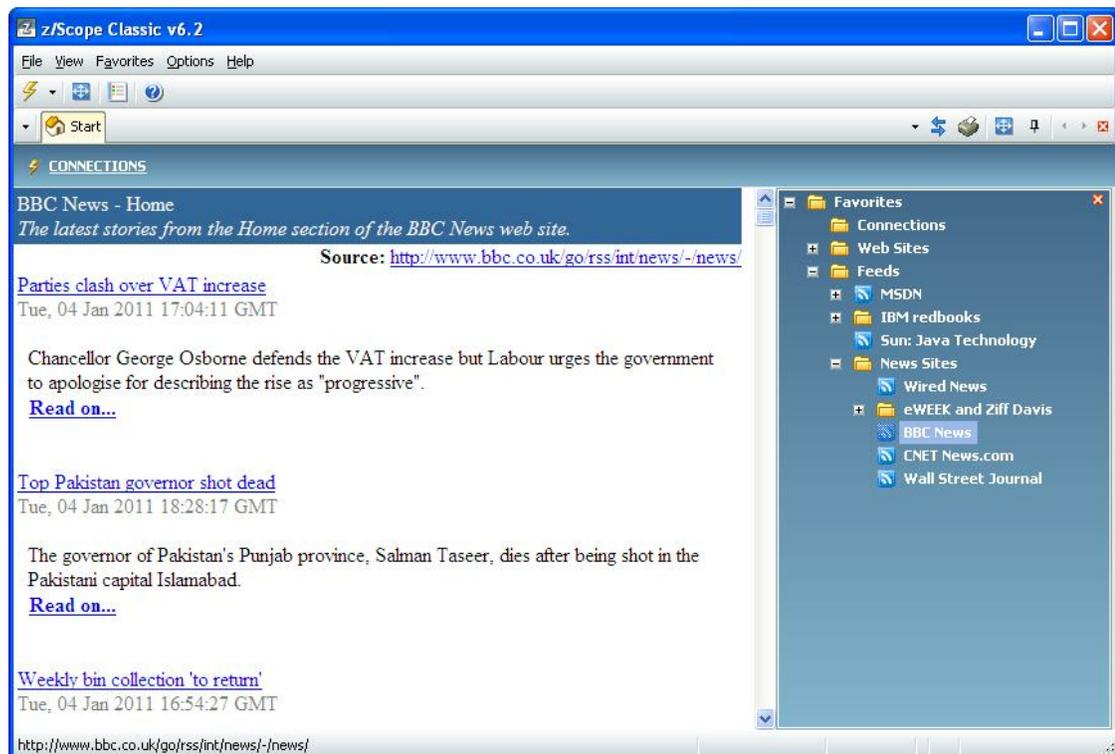
Z/Scope interface has four different but at the same time fully integrated workspaces. You can instantly switch between workspaces using the [Tab Selection Bar](#).

Each of the workspaces will be explained in detail throughout the following sections:

- [Start Page](#)
- [Telnet Workspace](#)
 - [Printer](#) (only Available for z/Scope Classic and Web-to-Host)
- [FTP Workspace](#)
- [Web Browser Workspace](#)

3.1.1 Start Page

The Start Page provides a convenient and quick access to Connections, Web Sites and Feeds.



The start page allows you to visualize:

- [The Favorites List](#)
- The list of configured connections.

- The RSS feeds accessed from the Favorites List.
- An RSS feed or web page of your choice when starting.

Also you can customize the Start Page's colors and other settings. To learn how to do this visit [Configuring the Start Page](#).

3.1.2 Tab Selection Bar

The Tab Selection Bar provides a convenient and organized way to access and manage all open tabs. It consists of two elements:



1. Open Workspaces Tabs

Provides quick one-click access to all currently open workspaces (Telnet, FTP sessions, and web pages). To switch between tabs, just click on the corresponding tab; the appropriate Menu and Toolbar options will be dynamically loaded according to the workspace invoked by the tab you selected. The Open Workspaces tab also includes:

-  **Start**
 This is the Start Page Tab. If you don't see it among the Open Workspaces Tabs, you can activate it from the 'View' Menu in the Main Menu. To learn more about the Start Page visit the [Start Page](#) section.
-  **Tiled View**
 Switches to tiled view. This button appears in the Open Workspaces Tab whenever there is more than one Telnet connection.
-  **Favorites**
 Quickly [Access](#), [Organize](#) and [Add Favorites](#).

2. Tab Navigation Buttons

These are useful controls that help you navigate through tabs:

-  **Switch**
 Click on this button to display a list of the current tabs.
-  **Show Printer Queue**
 Click on this button to open the printer panel.

-  Full Screen
Click on this button to enter Full Screen Mode.
-  Full Client
Click on this button while on Full Screen Mode to pin the toolbars.
-  Previous
Click on this button to switch to the previous tab.
-  Next
Click on this button to switch to the next tab.
-  Close
Click on this button to close the active tab.

3.1.3 Tiled View

The tiled view is a feature that will allow you to visualize more than one emulation screen at a time.

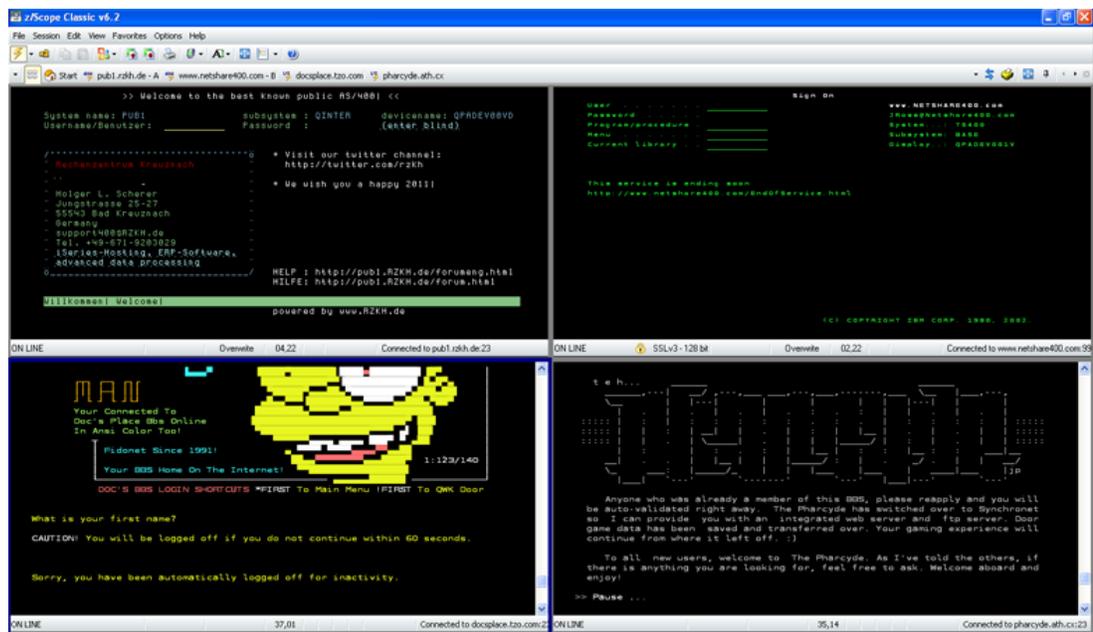
This feature can be activated whenever you have more than one emulation running at the same time.



FTP, web browser's tabs and the Start Page are not included for this count nor shown in the tiled view.

When the feature is available, you will see this  button appear as a new tab in the [Open Workspaces Tab](#). By pressing it, the emulation windows will rearrange in a tile pattern.

This is an example on how z/Scope's screen looks when visualizing tiled emulations:



ⓘ Please note that only the first four emulations on the toolbar will be shown in the tiled view.

In the tab selection bar, the tiled view option tab appears as selected. The active emulation will be framed in blue. You can work on the active emulation normally while seeing the progress on other screens.

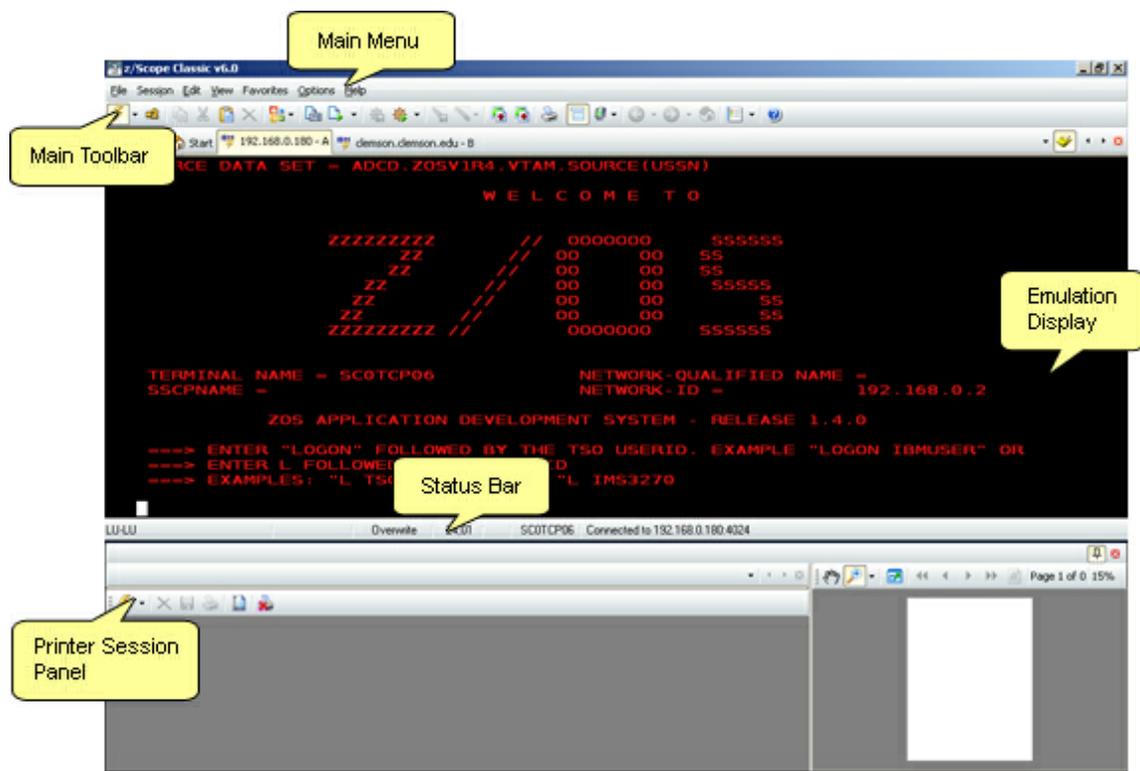
In order to switch between tiled emulations without leaving the tiled view, you can click on any of them with the mouse or simply press `ctrl + tab`.

You can go back to regular view just by clicking on any tab to see it full sized.

3.1.4 Telnet Workspace

The Telnet Workspace consists of the following main elements. Each one of them is briefly described in the following sections:

- [Main Menu](#)
- [Main Toolbar](#)
- [Emulation Display](#)
- [Printer Session Panel](#) (not available for Express editions)
- [Status Bar](#)

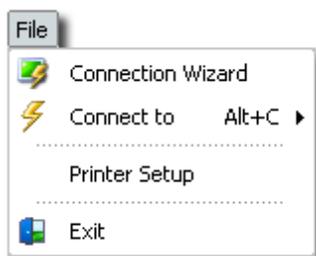


3.1.4.1 Main Menu

The Main Menu of the Telnet workspace consists in seven sub-menus. Each one of them will be covered in this topic.

File Session Edit View Favorites Options Help

File menu



Connection Wizard
Launches the Connection Wizard. See Setting up a Connection.

Connect to
Allows you to easily start a session for available connections.

Printer Setup
Displays the Printer Parameters dialog.

Exit
Closes z/Scope.

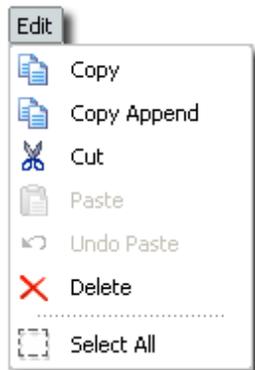
Session menu



Connect/Disconnect
Connects or disconnects the active connection.

Save Ebcdic Table
Use this option to save your current conversion table to a file. See [Modifying the Character Table](#).

Edit menu



Copy
Use this option to copy the selected text into the clipboard buffer.

Copy Append
Use this option to copy the selected text and append it to the text in the clipboard.

Cut
Use this option to copy the selected text into the clipboard buffer and delete the selected text, all in a single step.

! Only text in editable text fields will be deleted.

Paste

Use this option to paste text from the clipboard into the screen at the cursor position.

Undo Paste

Use this option to undo the last paste action.

Delete

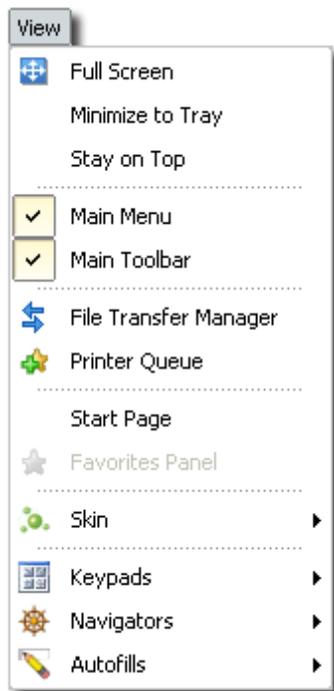
Use this option to delete the selected text.

! Only text in editable text fields will be deleted.

Select All

Use this option to automatically select the entire screen contents.

View menu



Full Screen

Check this option to enter Full Screen Mode.

Minimize to Tray

Checking this option will send the application to the System Tray when minimized, instead of the TaskBar.

Stay on Top

Checking this option will keep z/Scope on top of other Windows applications.

Main Menu

Uncheck this option to hide the Main Menu. To bring the Main Menu back, right click on any tab and check this option again.

Main Toolbar.

Uncheck or check this option to hide or see the Main Toolbar.

File Transfer Manager

Press this option to access the File Transfer Manager.

Printer Queue

Press this option to display the Printer Panel.

Start Page

This option makes the Start Page tab visible.

Favorites Panel

This option makes the Favorites Panel visible in the Start Page tab.

Skin

Allows you to choose different [Skins](#) for the application.

Keypads

Choose from a list of available [Keypads](#) the ones you want to see.

Navigators

Choose from a list of available [Navigators](#) the one you would like to apply.

Autofills

Choose from a list of available [Auto-Fills](#) the one you would like to apply.

Favorites menu



Add to Favorites

Adds current connection, web site or feed to the Favorites list. See [Building Your Favorites List](#) for more information.

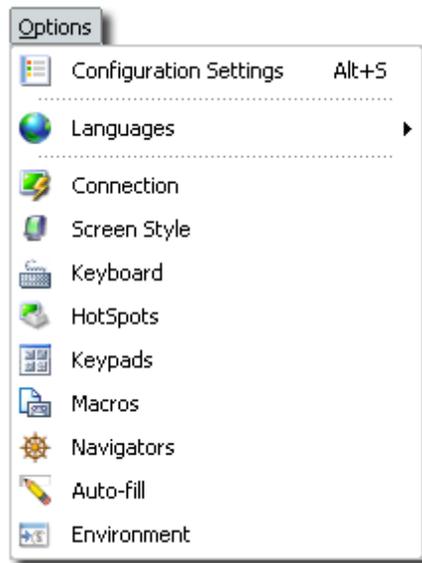
Organize Favorites

Opens the 'Organize Favorites dialog'. See [Organizing Your Favorites List](#) for more

information.

.....
Also this menu displays a list of existing Favorite feeds, connections and web sites organized in folders.

Options menu



Configuration Settings
Opens the 'Settings' dialog.

Languages
Allows you to choose different languages for the application.

Connection
Opens the [Connections Settings](#) for the current connection.

Screen Style
Opens the [Screen Style Settings](#) for the current Screen Style.

Keyboard
Opens the [Keyboard Settings](#).

HotSpots
Opens the [HotSpots Settings](#).

Keypads
Opens the [Keypads Settings](#).

 This option is not available for UNIX/VT connections.

Macros

Opens the [Macros Settings](#).

 This option is not available for UNIX/VT connections.

Navigators

Opens the [Navigators Settings](#).

 This option is not available for UNIX/VT connections.

Auto-fill

Opens the [Auto-fills Settings](#).

 This option is not available for UNIX/VT connections.

Environment

Opens the [Environment Settings](#).

Host Character Set

Allows you to choose different characters to display the current emulation. You can also configure this option by default for a connection in [Unix connections - Host Tab](#).

 This option is only available for UNIX/VT connections.

Help menu



Help Topics

Use this option to access this help file.

About

Shows information about z/Scope.

Related Topics

[Main toolbar](#)

3.1.4.2 Main Toolbar

The Main Toolbar of the Telnet workspace consists of the following options. Each one of them will be detailed in this topic.

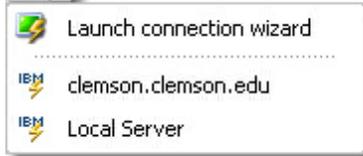


Connections section



Connect/Disconnect

Use this button to connect or disconnect the active connection.



Click on the arrow to unfold a list of all existing connections. Select the 'Launch Connection Wizard' option to create a new connection.



Duplicate active session

Use this button to open another session of the current connection so you can browse them independently.

Edit section



Copy

Use this option to copy the selected text into the clipboard buffer and delete the selected text, all in a single step.



Cut

Use this option to copy the selected text into the clipboard buffer.



Paste

Use this option to paste text from the clipboard into the screen at the cursor position.



Undo Paste

Use this option to undo the last Paste operation.



Delete

Use this button to delete the selected text.

Office Integration



Office Integration

Select a Microsoft Office application from the list to export the selected text area. See the topic [MS Office Integration](#) for more information.

Macros section



Start Macro Recording

Use this button to start recording a [macro](#).



Once pressed, this button will be replaced by the 'Stop Macro Recording' button.



Stop Macro Recording

Use this button to finish recording a [macro](#). A dialog box will appear asking you for a macro name in order to save it.



Once pressed, this button will be replaced by the 'Start Macro Recording' button.



Play Macro

Use this button to play a previously recorded macro. A list of all available macros will be unfolded. From this list you can also select the 'Edit' option which will launch the 'Active Script' dialog, allowing you to [edit](#) and [debug](#) macros.



Once pressed, this button will be replaced by the 'Stop Macro Reproduction' button.



Stop Macro Reproduction

Use this button to interrupt the macro reproduction.



Once pressed, this button will be replaced by the 'Play Macro' button.

Navigators section



Navigator Wizard

Use this button to launch the [Navigator Wizard](#).



Run Navigator

Select a [Navigator](#) to be run, from a list of all available Navigators for the current screen.

Auto-fills section



Auto-fill Wizard

Use this button to launch the [Auto-fill Wizard](#).



Run Auto-fill

Select an [Auto-fill](#) to be run, from a list of all available Auto-fills for the current screen.

File Transfer section



IND\$FILE File Upload

Use this button to upload a file from your PC to the remote computer

using either the IND\$FILE, KERMIT, XMODE, ZMODE, YMODE or File Transfer (FTP) Protocol. The File Transfer Dialog will pop up with the transfer direction set to 'Send'.



IND\$FILE File Download

Use this button to download a file to your PC from the remote computer using either the IND\$FILE, KERMIT, XMODE, ZMODE, YMODE or File Transfer (FTP) Protocol. The File Transfer Dialog will pop up with the transfer direction set to 'Receive'.

Print Screen



Print Screen

Use this button to send a hard copy of the emulation screen to the printer.

Miscellaneous section



Auto-Suggestion

Use this button to turn the [Auto-Suggestion](#) feature on/off.



Screen Styles

Use this button to change the [Screen Styles](#) for the current emulation.

Navigation section



Previous Screen

Use this button to go to the previous screen of the current emulation. Click on the arrow to unfold the [Screen History](#) thumbnail panel.



Next Screen

Use this button to go to the next screen of the current emulation. Click on the arrow to unfold the [Screen History](#) thumbnail panel.



Return to Active Screen

Use this button to return to the active screen after consulting a screen from the [Screen History](#).

Full Screen



Full Screen

Press this button to enter Full Screen Mode.

Settings



Settings

Use this button to display the general 'Settings' dialog.

Help



Help

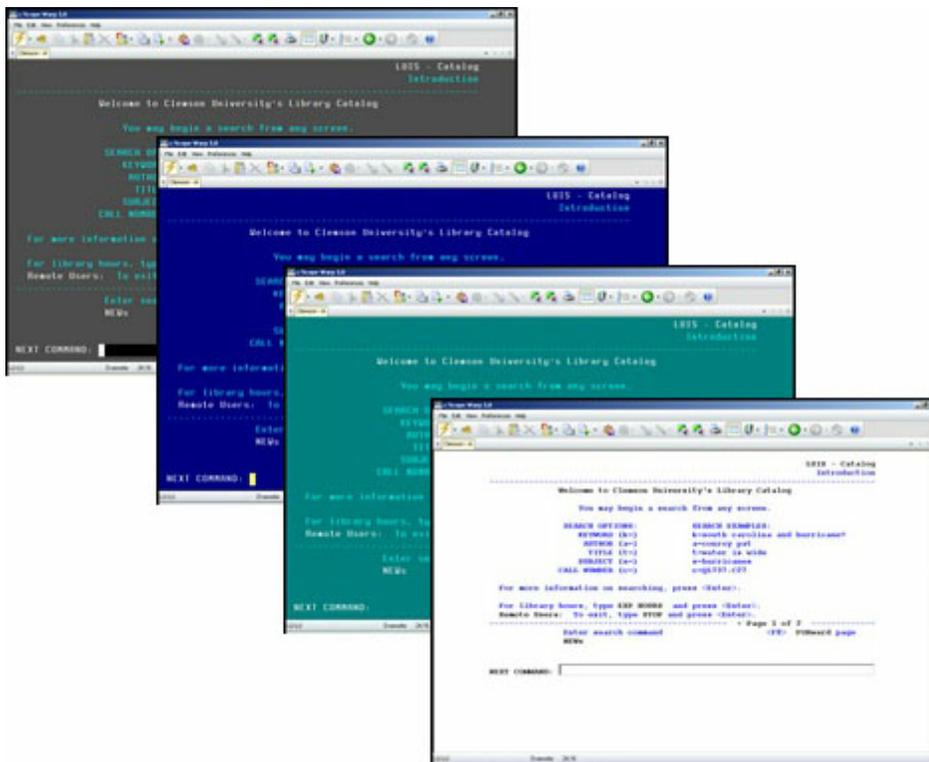
Use this button to get access to this help file.

Related Topics

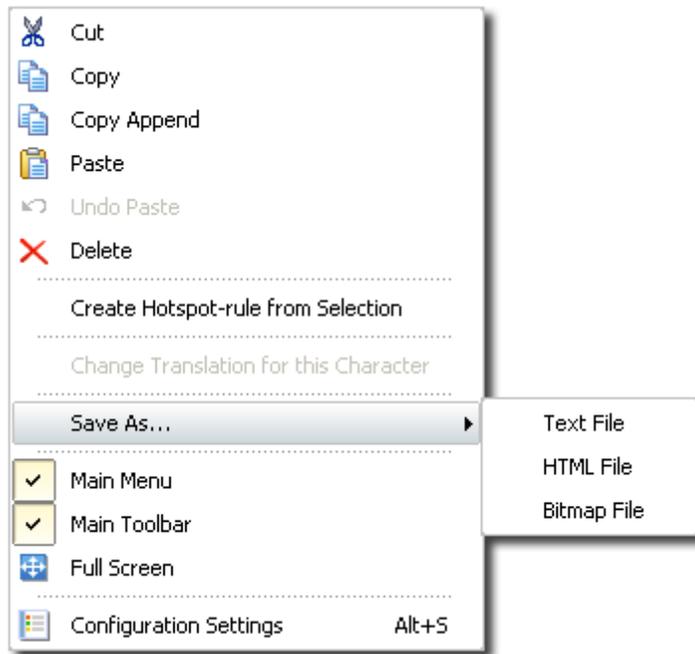
[Main Menu](#)

3.1.4.3 Emulation Display

This is where the terminal emulation actually takes place. As you can see in the screenshots below, the appearance of the Emulation Display can be fully customized by changing the font type, size and colors. You can also define sets of attributes and save them to the [Screen Styles](#) library, so they will be ready to be applied later to any session you want.



When you right click on the Emulation Display, a context menu will be displayed where you can have instant access to most commonly used edit options. These options will be covered below.



Cut

Use this option to copy the selected text into the clipboard buffer and delete the selected text, all in a single step.

 Only text in editable text fields will be deleted.

Copy

Use this option to copy the selected text into the clipboard buffer.

Copy Append

Use this option to copy the selected text and append it to the text in the clipboard.

Paste

Use this option to paste text from the clipboard into the screen at the cursor position.

Undo Paste

Use this option to undo the last paste action.

Delete

Use this option to delete the selected text.

 Only text in editable text fields will be deleted.

Create HotSpot rule from selection

Use this option to [create a new HotSpot](#) using the selected text as keyword.

Change Translation for This Character

Use this option to modify the EBCDIC to ASCII conversion for a particular character.

See [Modifying the Character Table](#).

Save As...

Use this option to save the selected text or the full screen into one of the following formats: Text File (.txt), HTML File (.html) and Bitmap File (.bmp). A dialog will be displayed to provide the name and path of the file to be saved.

Main Menu

Uncheck this option to hide the Main Menu. To bring the Main Menu back, right click on any tab and check this option again.

Main Toolbar.

Uncheck or check this option to hide or see the Main Toolbar.

Full Screen

Check this option to enter Full Screen Mode.

Configuration Settings

Access z/Scope Settings.

Related Topics

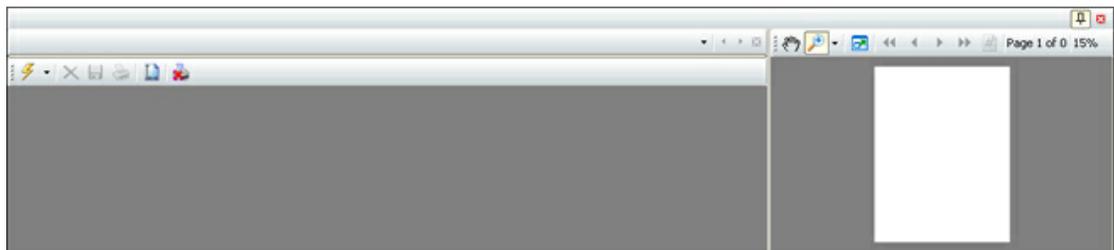
[Using Screen Styles](#)

3.1.4.4 Printer Session Panel

Z/Scope allows Mainframe's print-job redirection to local or network printers by implementing SNA Character String data stream over TN3270E or TN5250E standard protocols.

It emulates IBM-3812-1 and IBM-3287-1 printer models in direct-printing, print-to-file or print-preview mode, as well as providing full support for Host Print Transform, now all integrated into [z/Scope's tabbed interface](#).

Printer sessions are managed from the Printer Session Panel, which can be shown at any time by clicking on the 'Printer' icon  located to the right of the [Tab Selection Bar](#).



Click on the  button to hide the Printer Session Panel.

Click on the  button to lock the Printer Session Panel so it won't automatically hide

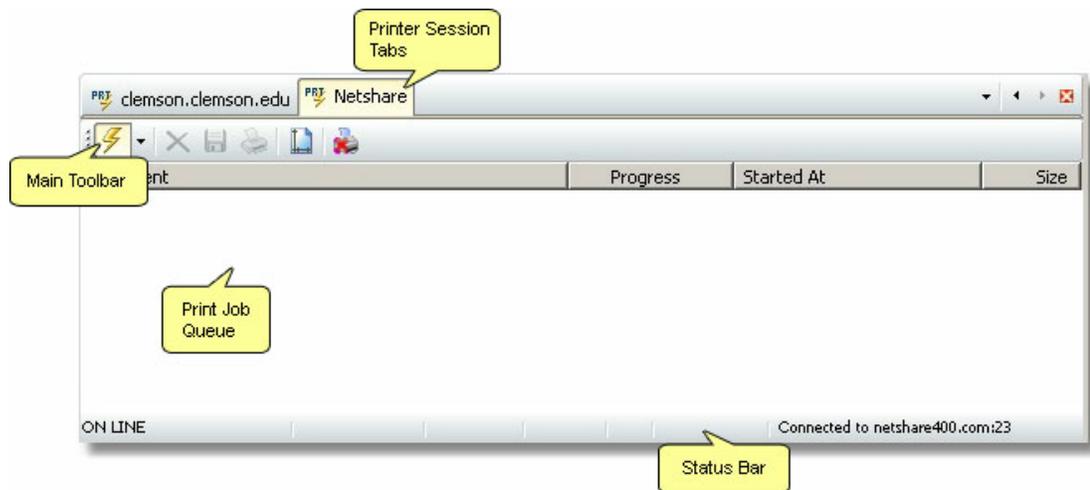
when idle.

To find out more about each of the components of the Printer Sessions Panel, select one of the following links:

- [Print Job Queue](#)
- [Print Preview Dialog](#)
- [Printer Parameters Dialog](#)

3.1.4.4.1 Print Job Queue

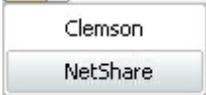
The Print Job Queue interface consists of four main elements. Each one of them is briefly described in this topic.



Printer Session Tabs

The Printer Session Tab Selection Toolbar is intended to quick access to all active printer sessions. Consult the main [Tab Selection Bar](#) topic for more information on the tabbed interface functioning.

Main Printer Toolbar

-  Connect/Disconnect
Use this button to connect or disconnect the active printer session.
- 

 Click on the arrow to unfold a list of all existing printer connections.
-  Delete
Use this button to delete the selected print job from the Print Job Queue.



Save

Use this button to save the current print job queue to a file.



Print

Use this button to print the selected print job.



Page Setup

Use this button to display the [Printer Parameters](#) dialog.



Hold Printing

Use this button to pause printing.

Print Job Queue

The Print Job Queue shows information about each print job that was initialized within the current connection.

Document column

The Document column shows the name of the print job.

Progress column

The Progress column shows the current status of the print job.

Started At column

The Started At column shows the date and time at which the print job was started.

Size column

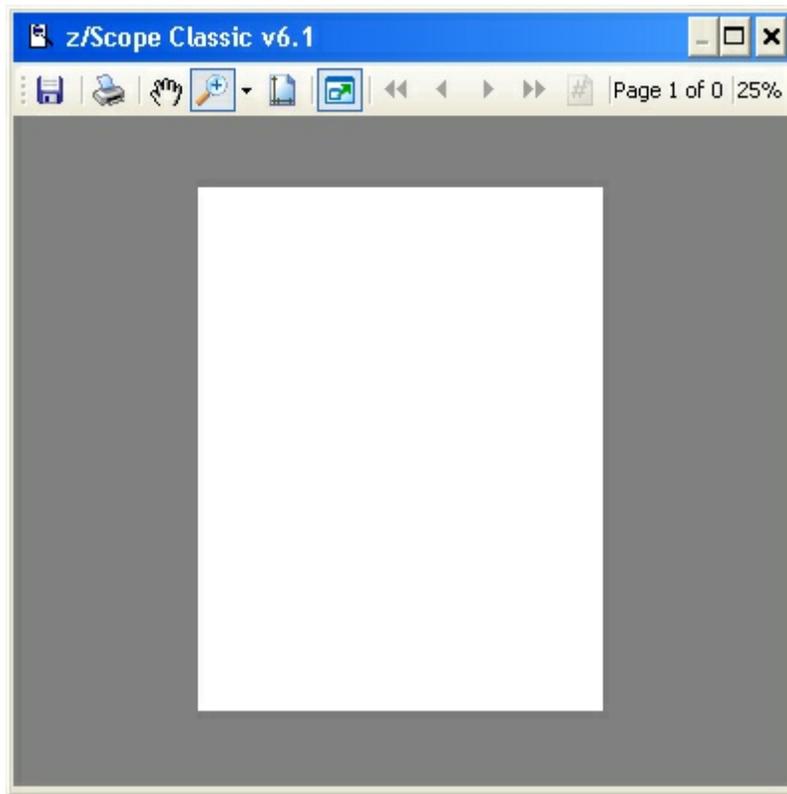
The Size column shows the size of the print job.

Status Bar

The Status Bar gives you real-time information about the status of several relevant parameters of the current Print Job. The information displayed in the Status Bar is visible to the user at all times, thus providing a constant update on several useful parameters and settings.

3.1.4.4.2 Print Preview Dialog

The Print Preview Dialog interface consists in two main elements: The Print Preview Toolbar and the Page Preview.



Print Preview Toolbar

The Print preview toolbar presents a series of buttons that will be described below. This toolbar also displays the page number that we are previewing, the total of pages in the document and the current level of zoom in the preview.

-  Save
Use this button to save the selected print job to a file.
-  Print
Use this button to start printing the selected print job.
-  Hand Tool
Use this button to pan the current page.
-  Zoom Tool
Use this button to set the desired level of zoom.
-  Dock/Undock
Use this button to dock/undock the Print Preview Dialog.
-  First Page
Use this button to move to the first page of the selected document.

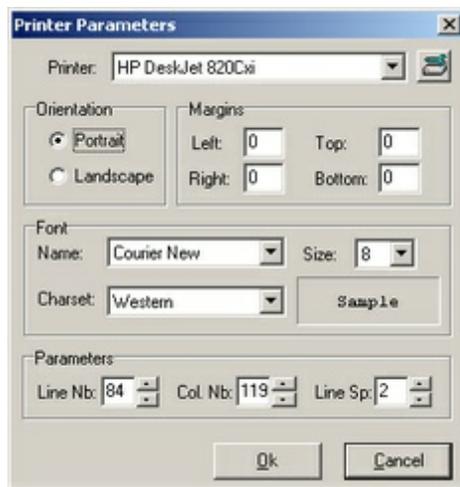
-  Previous Page
Use this button to move to the previous page of the selected document.
-  Next Page
Use this button to move to the next page of the selected document.
-  Last Page
Use this button to move to the last page of the selected document.
-  Page Number
Use this button to enter the page number you want to move to.

Page Preview

The Page Preview area shows a page preview of the selected document.

3.1.4.4.3 Printer Parameters Dialog

The 'Printer Parameters' dialog has the following parameters:



Printer

This list contains all available printers in your local computer. Use the 'Print Setup' button to the right of the 'Printer' field to open Windows' common dialog for configuring printing options.

Orientation

Choose between portrait or landscape paper orientation.

Margins

In this fields you must specify the left, right, top and bottom margins in centimeters (1 inch = 2.54 cm).

Font

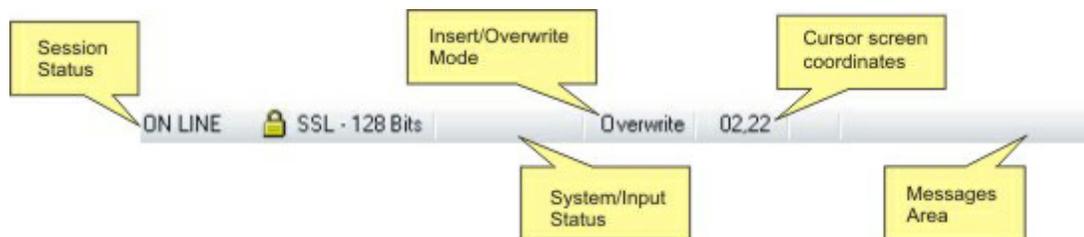
This list contains all installed fonts. You can also set the font size and character set. The 'Font Preview' area shows how the selected font will look like according to the chosen parameters.

Parameters

Set the number of lines and columns for the document, and the line spacing.

3.1.4.5 Status Bar

The Status Bar gives you real-time information about the status of several relevant parameters of the current Workspace. The information displayed in the Status Bar is visible to the user at all time, thus providing a constant update on several useful parameters and settings.



The status bar is divided into five areas, as described below:

Session Status

Shows information about the status of the current session. When working with secure protocols such as SSL or TLS, information on those features is also shown here.

System/Input Status

Shows the current status the system, such as wait states. This panel also indicates whether the terminal is currently accepting user input or not.

Insert/Overwrite Mode

Indicates text insertion mode. In 'Overwrite' mode, new characters will overwrite existing ones. In 'Insert' mode, new characters will be inserted in the selected position without overwriting the existing ones. To switch between these modes, use the 'Insert' key in your keyboard.

Cursor Screen-Coordinates

These are the actual screen coordinates of the cursor. The first number refers to the current row, and the second one to the current column.

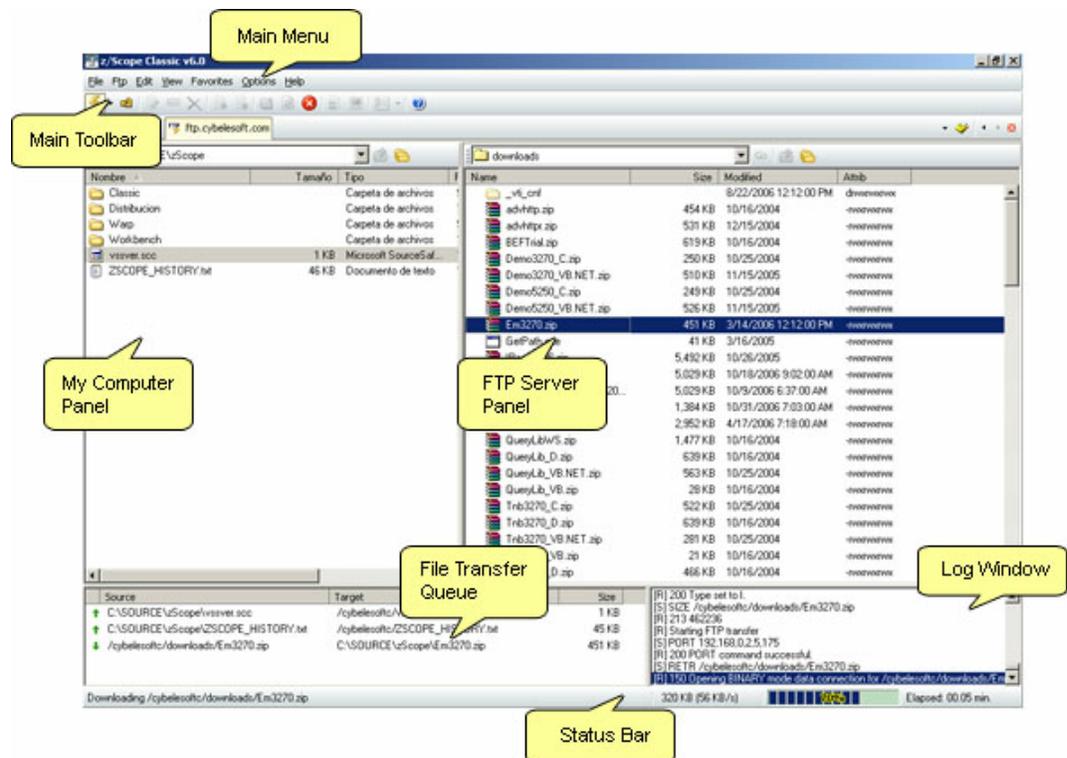
Messages Area

This area is used by z/Scope to return additional information to the user.

3.1.5 FTP Workspace

Z/Scope's FTP workspace consists of the following main elements. Each one of them will be covered in the following sections:

- [Main Menu](#)
- [Main Toolbar](#)
- [My Computer Panel](#)
- [FTP Server Panel](#)
- [File Transfer Queue](#)
- [Queue Item Edit](#)
- [Log Window](#)
- [Status Bar](#)



3.1.5.1 Main Menu

The Main Menu of the FTP workspace consists of seven sub-menus. Each one of them will be covered in this topic.

File Ftp Edit View Favorites Options Help

File menu



Connection Wizard

Launches the Connection Wizard. See Setting up a Connection.

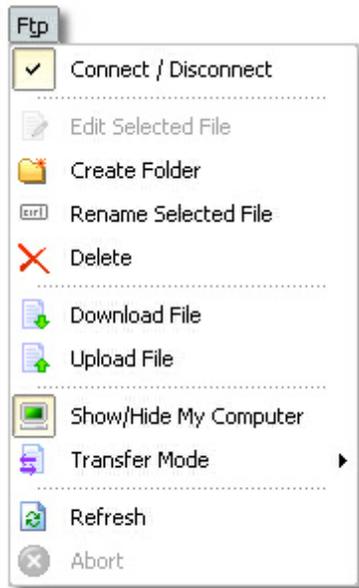
Connect to

Allows you to easily re-connect a recently used Connection.

Exit

Terminates z/Scope.

FTP menu



Connect/Disconnect

Connects or disconnects the active connection.

Edit Selected File

Use this option to open the selected file. You can also open a file for editing simply by double-clicking on the file.

Create Folder

Use this option to create a new folder in the current location.

Rename Selected File

Use this option to rename the selected file.

Delete

Use this option to delete the selected file.

Download File

Use this option to transfer the selected file from the remote computer to your PC.

Upload File

Use this option to transfer the selected file from your PC to the remote computer.

Show/Hide My Computer

Use this option to show or hide the 'My Computer' panel.

Mode

Use this option to select the transmission mode. Available options are 'ASCII', 'Binary' and 'Auto'.

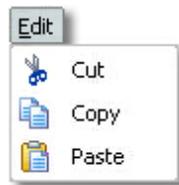
Refresh

Use this option to update the contents of the folders.

Abort

Use this option to stop a file transfer.

Edit menu



Cut

Use this option to copy the selected file into the clipboard buffer and delete the selected file, all in a single step.

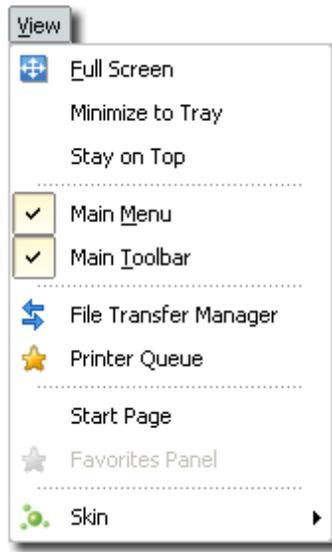
Copy

Use this option to copy the selected file into the clipboard buffer.

Paste

Use this option to paste a file from the clipboard into the current location.

View menu



Minimize to Tray

Checking this option will send the application to the System Tray when minimized, instead of the TaskBar.

Stay on Top

Checking this option will keep z/Scope on top of other Windows applications.

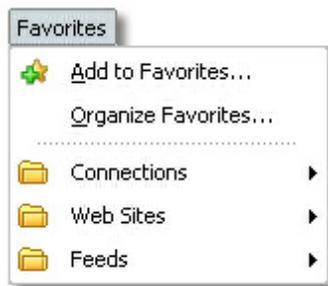
Start Page

This option opens the Start Page tab.

Skin

Allows you to choose different [Skins](#) for the application.

Favorites menu



Add to Favorites

Adds current connection, web site or feed to the Favorites list. See [Building Your Favorites List](#) for more information.

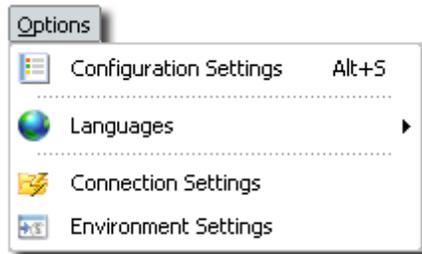
Organize Favorites

Opens the 'Organize Favorites' dialog. See [Organizing Your Favorites List](#) for more

information.

Also this menu displays a list of existing Favorite feeds, connections and web sites organized in folders.

Options menu



Configuration Settings
Opens the Settings.

Languages
Allows you to choose different languages for the application.

Connection Settings
Opens the [Connections Settings for the FTP connection](#).

Environment Settings
Opens the [Environment Settings](#).

Help menu



Help Topics
Use this option to access this help.

About
Shows information about z/Scope.

Related Topics

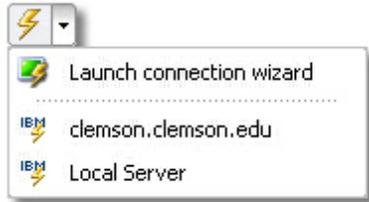
[FTP Main Toolbar](#)

3.1.5.2 Main Toolbar

The Main Toolbar of the FTP Client workspace consists of the following options. Each one of them will be detailed in this topic.



Connections section



Connect/Disconnect

Use this button to connect or disconnect the active connection. Click on the arrow to unfold a list of all defined connections. Select the 'Connection Wizard' option to create a new connection.



Duplicate active session

Use this button to create a clone of the current connection so you can browse them independently.

Edit Section



Edit

Use this button to open the selected file with your system's default text editor.



Rename

Use this button to rename the selected file.



Delete

Use this button to delete the selected file.

File Transfer Section



File Download

Use this button to transfer the selected file from the remote computer to your PC.



File Upload

Use this button to transfer the selected file from your PC to the remote computer.



New Folder

Use this button to create a new folder in the remote host.



Refresh

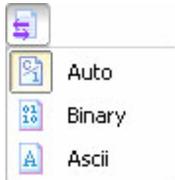
Use this button to update the contents of the folders.



Abort

Use this button to stop a file transfer.

Settings Section



Mode

Use this button to select the transmission mode. Available options are 'ASCII', 'Binary' and 'Auto'.



Show/Hide My Computer

Use this button to show or hide the 'My Computer' panel.



Settings

Use this button to display the general 'Settings' dialog.



Help

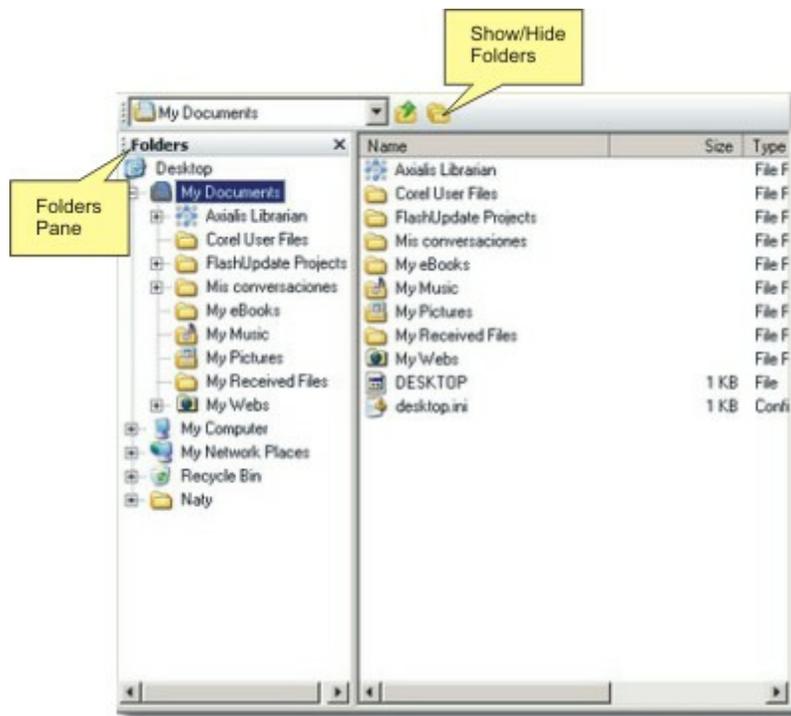
Use this button to get access to this help.

Related Topics

[Main Menu](#)

3.1.5.3 My Computer Panel

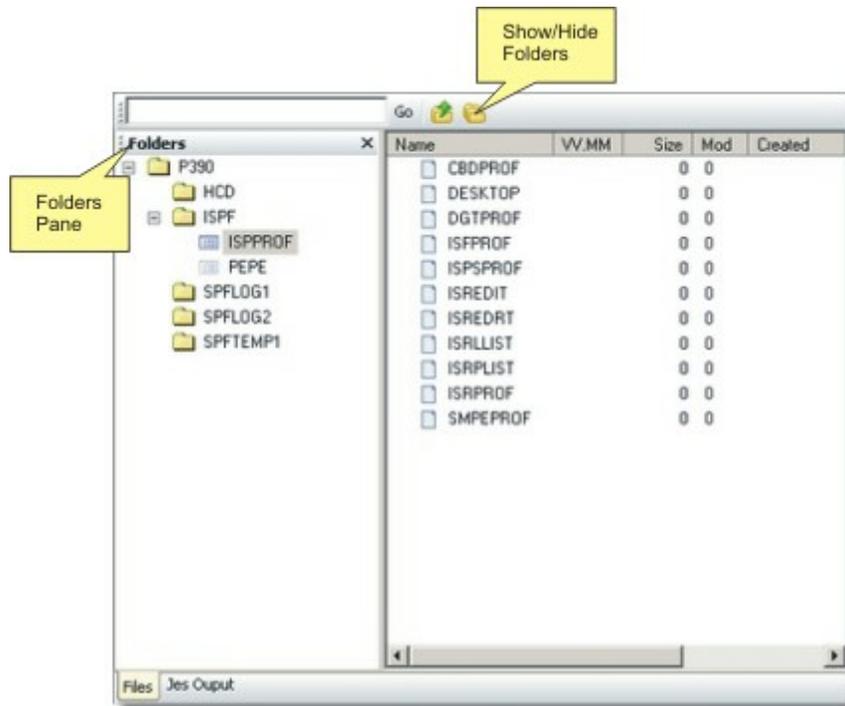
In the 'My Computer' panel you can browse through the files stored in your local computer. You can browse for the files you need to upload, and when you start a file download, the currently active folder in this panel will be used as the local destination folder for the file.



The Folders Pane can be hidden by clicking on the 'Show/Hide Folders' button.

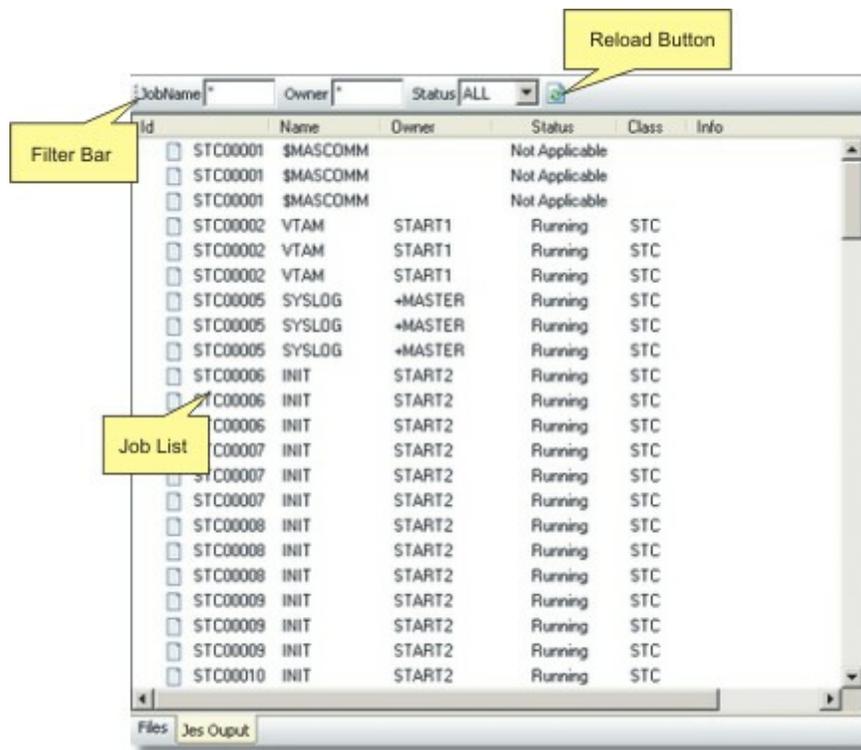
3.1.5.4 FTP Server Panel

In the 'FTP Server' panel you can browse through the files stored in the remote computer. When you start a file upload, the currently active folder in this panel will be used as the remote destination folder for the file.



The Folders Panel can be hidden by clicking on the 'Hide/Show Folders' button.

When connecting to MVS hosts, the 'Jes Output' tab will appear at the bottom of the panel. Click on it to display Jes Output results.



You can filter the Jes Output using the 'Filter Bar'. The information displayed can be filtered by job name, owner or status. Once you have entered the filters, click on the 'Reload' button to apply them.

To display all the results, clear both 'JobName' and 'Owner' fields, select 'ALL' from the 'Status' list, and then click on the 'Reload' button.

3.1.5.5 File Transfer Queue

Every time you start a file transfer using the 'Upload/Download' buttons in the FTP toolbar or dragging the file to the destination folder, the file will be added to the 'File Transfer Queue' ready to be transferred right away. You can also drag and drop files from the 'My computer' or 'FTP Server' panels to the queue if you don't want the transfer to start immediately. You can later choose when to start the transfer as explained later.

The screenshot shows a table with three columns: Source, Target, and Size. The rows represent files in the queue. The first row is 'P390.SPFTTEMP1.CNTL' with a size of 1 KB. The second row is 'C:\WINDOWS\explorer.exe' with a size of 973 KB. The third row is 'C:\WINDOWS\muninst.exe' with a size of 32 KB. Labels indicate 'Source File Name', 'File Size', 'Upload/Download indicator', and 'Target File Name'.

Source	Target	Size
↓ 'P390.SPFTTEMP1.CNTL'	C:\WINDOWS\system32\CNTL	1 KB
↑ C:\WINDOWS\explorer.exe	'P390.explorer'	973 KB
↑ C:\WINDOWS\muninst.exe	'P390.muninst'	32 KB

Upload/Download indicator

The arrows indicate if the file is marked for upload or for download.

Source File Name

This column shows the name and directory of the original file that is being transferred.

Target File Name

This column shows the name the file will be saved as and the directory in which it will be saved.

File Size

This column shows the size of the file.

If you right-click on an item in the File Transfer Queue, the following context menu will be displayed:

**Start**

Click on this option to start transferring a the selected file.

Start All

Click on this option to start the transfer of all the files in the queue.

Edit

Clicking on this option will open a the ['Queue Item Edit' dialog](#) to edit the transfer parameters

Delete

Clicking on this option will delete the selected queue entry.

Clear

Clicking on this option will clear the entire queue list.

Load From File

Use this option to load a previously saved queue list.

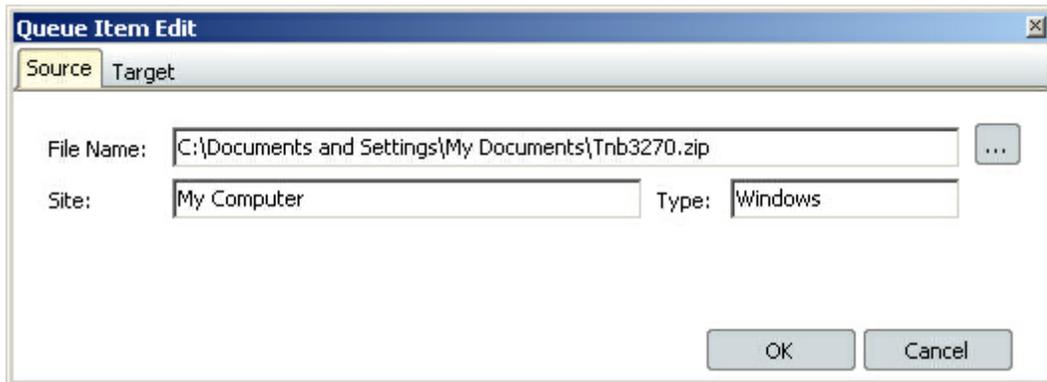
Save To File

Use this option to save the current queue list to a file.

3.1.5.6 Queue Item Edit

The Queue Item Edit dialog is used to configure the transfer parameters while a file is in the [File Transfer Queue](#) before the transfer begins.

In the source tab you will have the following parameters:



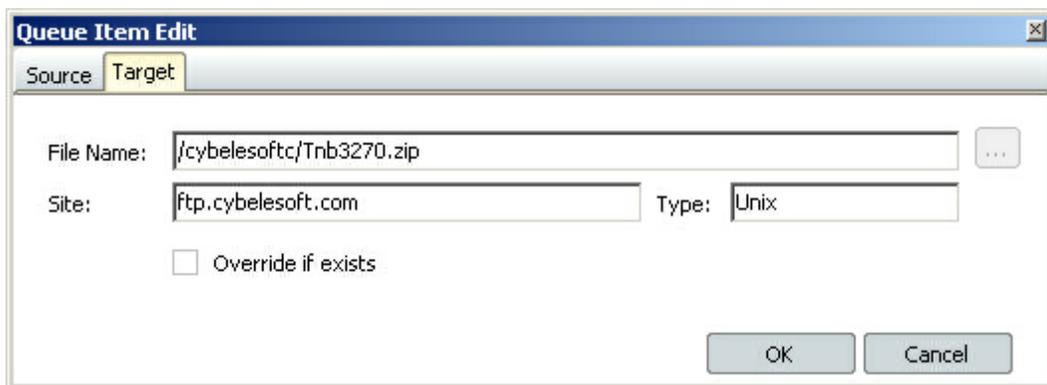
File Name

This is the name and directory of the file to be transferred. Modify this parameter to select a different file. You can type it or use the "." button to browse through the source folder.

Site and Type

These parameters reflect the current location and operating system of the above selected file.

In the Target tab you will have the following parameters:



File Name

In this field you specify the directory where the transferred file will be saved and the name it will take. Modify this parameter to select a different folder or name for the transferred file to be saved.

Site and Type

These parameters reflect the current location and operating system of the file selected above.

Override if exists

This option considers the case that there is already a file with the same name in the specified directory. Mark it if you would like to overwrite the existing file with the new one. If unchecked, the system will ask you to decide in the moment.

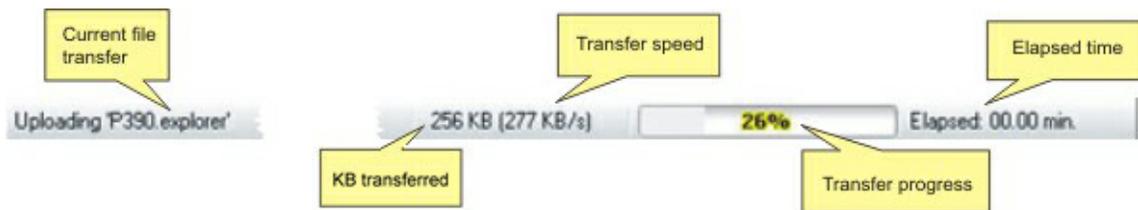
3.1.5.7 Log Window

The Log Window contains all the FTP commands sent to the FTP server by z/Scope, as well as any outcome information returned by the FTP server.

```
[R] Connecting to DNS=192.168.0.180 IP= PORT=21
[R] Connected
[R] 220 FTPD1 IBM FTP CS V1R4 at p390.svo.dlv.ibm.com, 15:19:17 on 2005-09-19.
Connection will close if idle for more than 5 minutes.
[R] 230 P390 is logged on. Working directory is "P390.".
[S] PWD
[R] 257 "P390." is working directory.
[S] TYPE A
[R] 200 Representation type is Ascii NonPrint
[S] PORT 192,168,0,100,4,146
[R] 200 Port request OK.
[S] LIST
[R] 125 List started OK
[R] 250 List completed successfully.
[S] TYPE I
[R] 200 Representation type is Image
[S] PORT 192,168,0,100,4,150
```

3.1.5.8 Status Bar

The Status Bar provides all the information concerning the file transfer operation that is currently in progress.



Current file transfer

Indicates the name of the file that is being transferred and if the operation is an upload or a download.

KB transferred

Dinamically shows the progress of the transfer in transferred KB from the current file.

Transfer speed

Shows transfer speed estimate.

Transfer progress

Shows the percentage of the file that has been transferred so far.

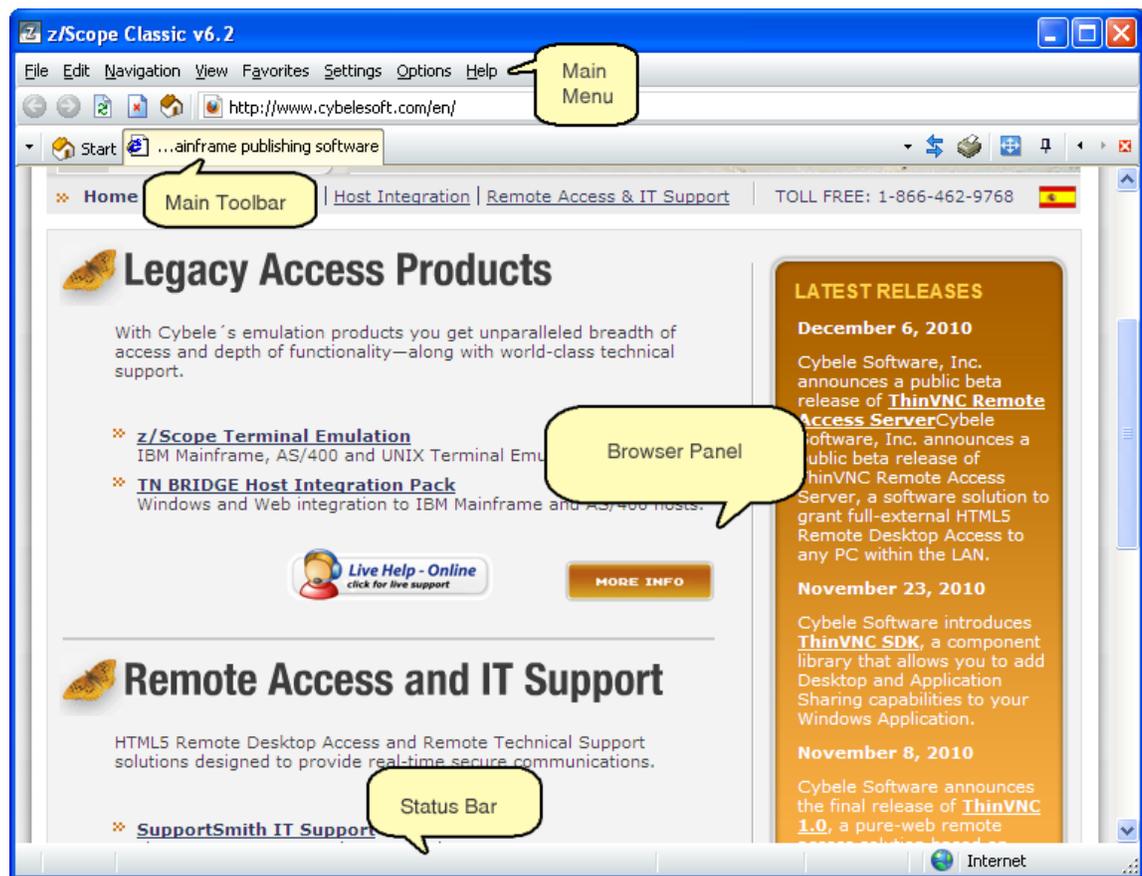
Elapsed time

Indicates the elapsed time since the operation was started.

3.1.6 Web Browser Workspace

Z/Scope's Web Browser workspace consists of the following main elements. Each one of them will be covered in the following sections:

- [Main Menu](#)
- [Main Toolbar](#)
- [Browser Panel](#)
- [Status Bar](#)

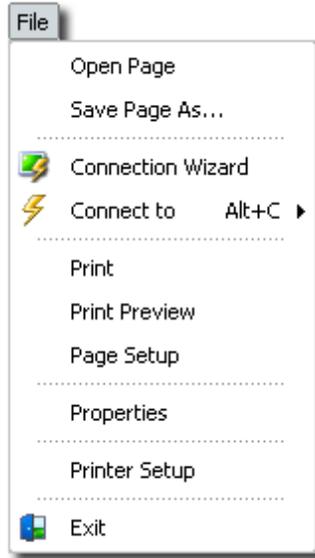


3.1.6.1 Main Menu

The Main Menu of the Web Browser workspace consists in eight sub-menus. Each one of them will be covered in this topic.

File Edit Navigation View Favorites Settings Options Help

File menu



Open Page

Opens a web page stored as a file.

Save Page As...

Allows to save the current web page to a file.

Connection Wizard

Invokes the 'Connection Wizard' dialog. See Setting up a Connection.

Connect to

Allows you to easily re-connect to a recently used Connection.

Print

Prints the current web page.

Print Preview

Allows you to preview the current web page before printing it.

Page Setup

Use this option to display the Internet Explorer page setup dialog.

Properties

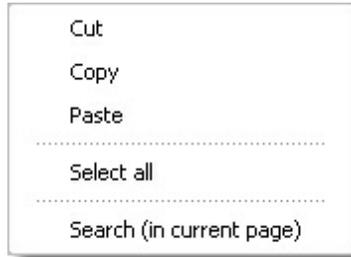
Opens the Internet Explorer Settings.

Printer Setup

Displays the Printer Parameters dialog.

Exit
Terminates z/Scope.

Edit menu



Cut
Use this option to copy the selected text into the clipboard buffer and delete the selected text, all in a single step.

Copy
Use this option to copy the selected text into the clipboard buffer.

Paste
Use this option to paste text from the clipboard into the web page at the cursor position.

Select All
Use this option to automatically select the entire web page contents.

Search (in current page)
Allows you to quickly find any text string in the current web page.

Navigation menu



Back
Goes to the previous page in the navigation sequence.

Forward
Goes to the next page in the navigation sequence.

Refresh

Refreshes all page contents from the current web site.

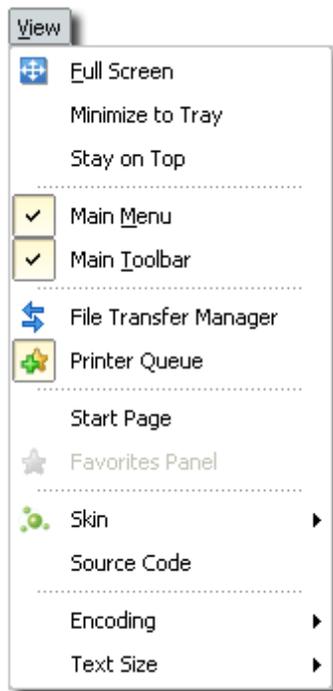
Abort

Stops loading the current page.

Home

Loads the Internet Explorer home page.

View menu



Full Screen

Check this option to enter Full Screen Mode.

Minimize to Tray

Checking this option will send the application to the System Tray when minimized, instead of the TaskBar.

Stay on Top

Checking this option will keep z/Scope on top of other Windows applications.

Main Menu

Uncheck this option to hide the Main Menu. To bring the Main Menu back, right click on any tab and check this option again.

Main Toolbar.

Uncheck or check this option to hide or see the Main Toolbar.

File Transfer Manager

Press this option to access the File Transfer Manager.

Printer Queue

Press this option to display the Printer Panel.

Start Page

This option makes the Start Page tab visible.

Favorites Panel

This option makes the Favorites Panel visible in the Start Page tab.

Skin

Allows you to choose different [Skins](#) for the application.

Source Code

Allows you to see the HTML source code of the page.

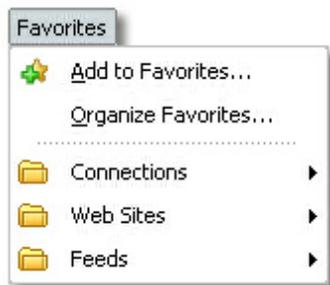
Encoding

Allows you to choose a different character set for the current page.

Text Size

Changes the font size of the current page.

Favorites menu



Add to Favorites

Adds current connection, web site or feed to the Favorites list. See [Building Your Favorites List](#) for more information.

Organize Favorites

Opens the 'Organize Favorites dialog'. See [Organizing Your Favorites List](#) for more information.

Also this menu displays a list of existing Favorites feeds, connections and web sites organized in folders.

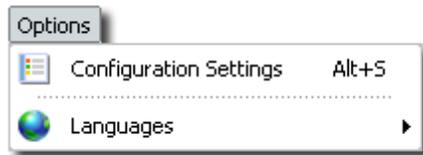
Settings menu



Internet Options
Opens the Internet Explorer Settings.

Search Engine
Allows you to choose different search engines to perform a search.

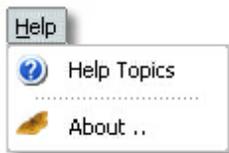
Options menu



Configuration Settings
Opens the 'Settings' dialog.

Languages
Allows you to choose different languages for the application.

Help menu



Help Topics
Use this option to access this help file.

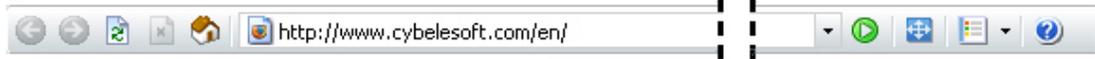
About
Shows information about z/Scope.

Related Topics

[Main Toolbar](#)

3.1.6.2 Main Toolbar

The Main Toolbar of the Web Browser workspace consists in the following options. Each one of them will be detailed in this topic.



Navigation section

-  **Back**
This button takes you to the previous page in the navigation sequence.
-  **Forward**
This button takes you to the next page in the navigation sequence.
-  **Refresh**
This button reloads all page contents from the current web site.
-  **Abort**
This button stops loading the current page.
-  **Home**
This button loads the Internet Explorer default home page.



Address Bar

Type the URL of the web page you want to load and then press enter or press the 'Go' button.

-  **Go**
This button loads the web page specified in the Address Bar.

Miscellaneous section

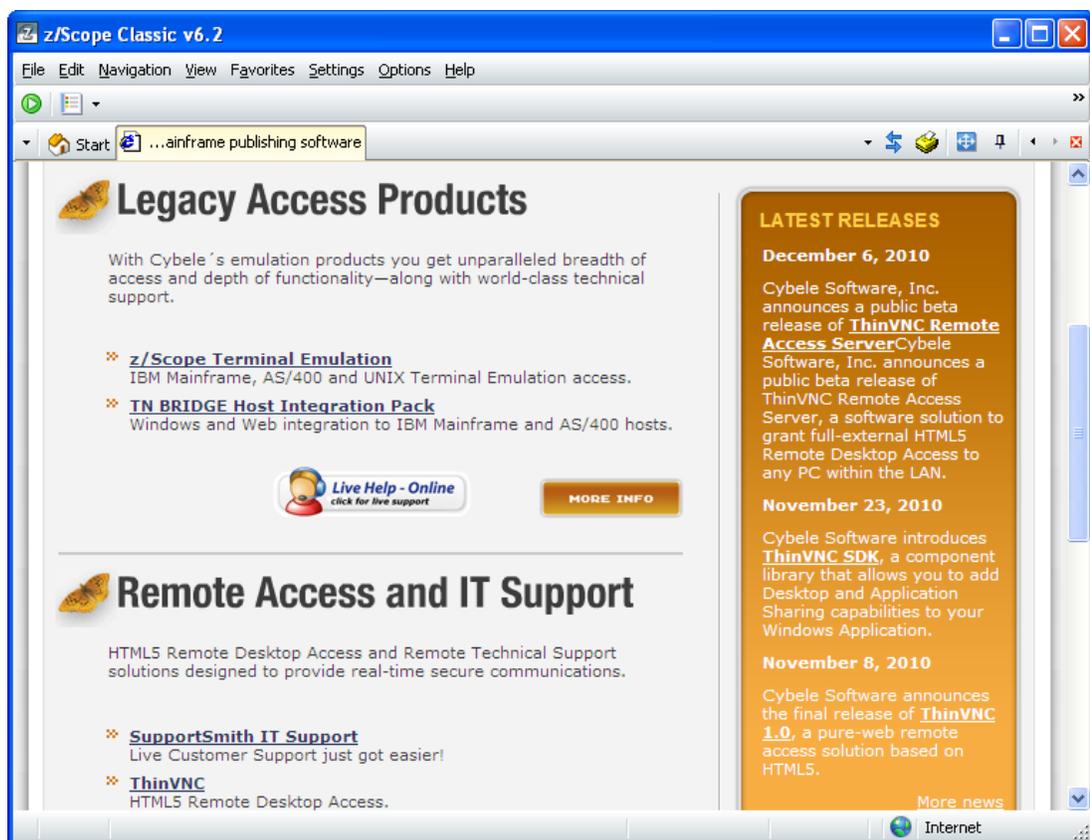
-  **Full Screen**
Press this button to enter Full Screen Mode.
-  **Settings**
Click on the 'Settings' button to access z/Scope Settings, or expand and choose:
Internet Options: To display Internet Explorer options or
Environment: To open z/Scope Environment Settings.
-  **Help**
Use this option to access this help file.

Related Topics

[Main Menu](#)

3.1.6.3 Browser Panel

The browser panel is the space where the web page is displayed. It also includes a tabbed interface to switch between different websites or other z/Scope workspaces. For more information on the tabbed interface, see the [Tab Selection Bar](#) section.



3.1.6.4 Status Bar

The Status Bar gives you real-time information about the status of several relevant parameters of the current Workspace. The information displayed in the Status Bar is visible to the user at all time, thus providing a constant update on several useful parameters and settings.

The Web Browser status bar provides information -according to your settings- about the web page reliability. Also it shows the loading process through a percentage bar and gives additional messages in specific situations.

3.2 Key Features

Z/Scope's features were specifically designed to improve your productivity and terminal emulation experience.

The following sections are intended to give you a first look at these features:

- [Auto-fills](#)
- [Auto-suggestion](#)
- [Favorites](#)
- [File Transfer](#)
- [HotSpots](#)
- [Keyboard Support](#)
- [Keypads](#)
- [Macros](#)
- [MS Office Integration](#)
- [Navigators](#)
- [OHIO](#)
- [HLLAPI](#)
- [Screen History](#)
- [Screen Styles](#)
- [Using z/Scope Skins](#)
- [Security and Encryption](#)

Copyright © 2002-2012, Cybele Software Inc. All rights reserved.

3.2.1 Auto-fills

The Auto-fill feature comes in handy when you have to complete many screen fields repeatedly. Auto-fills allow you to automatically fill-in one or more input fields with a simple mouse click.

To learn more about the Auto-fill feature, read the next sections:

- [Creating an Auto-fill](#)
- [Using Auto-fills](#)

Related Topics

[Auto-fill Settings](#)

3.2.1.1 Creating an Auto-Fill

Auto-fills are easily created through the Auto-fill Wizard.

First of all, type the text string you want to define as an Auto-fill into the corresponding input field and then press the Auto-fill Wizard button: 

After that, you will be presented with the Auto-fill Wizard welcome screen.

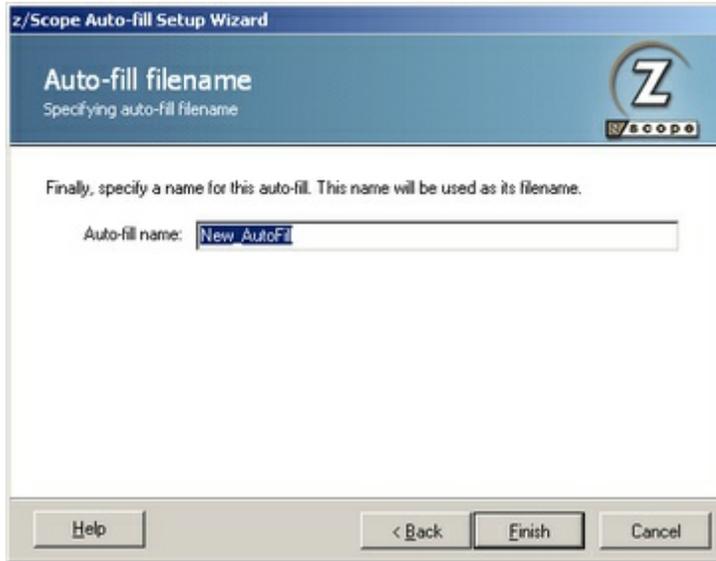


Click on 'Next' to go to the next step, in which you will have to specify the following parameters:



Auto-fill label
In this field you must enter a label for the Auto-fill button.

Click on 'Next' to go to the last step of the Auto-fill creation process.



Auto-fill name

In this field you must enter a file name for the Auto-Fill you are creating.

Click on 'Finish' to confirm the new Auto-fill creation. Otherwise, click on 'Cancel' to discard the whole procedure.

If the 'Auto-fill Wizard' is invoked from a password input field, the first dialog that you will see after the Welcome dialog will be the 'Password field handling' dialog, in which you can choose one the following options:



Encrypt

Select this option if you want the text string to be stored encrypted. This is the same as setting the 'Switch to encrypted field' option in the ['Screen Fields' tab of the 'Auto-Fills Settings' dialog](#).

Prompt for password

Select this option if you want to be prompted for the password of filling it automatically.

Ignore

Select this option if you don't want the Auto-fill to complete this field. Every other field will be completed, except for the password field that will be left to be filled manually by the user.

To learn how to use Auto-fills, see the next topic:

- [Using an Auto-fill](#)

Related Topics

[Auto-Fill Settings](#)

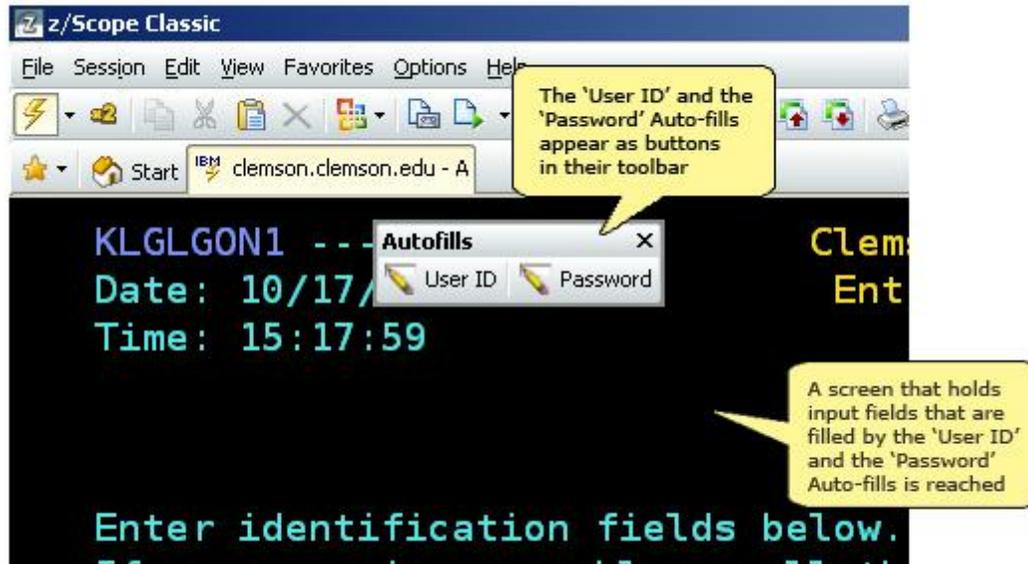
3.2.1.2 Using an Auto-Fill

There are two ways in which an Auto-fill can be invoked:

1. From a button in a user-defined toolbar

If the Auto-fill is configured to work as a button in a toolbar (configure this option in the

[General Tab of the 'Auto-fill Settings' dialog](#)), it will automatically appear in its corresponding toolbar at the top of the emulation display (as shown in the screenshot below) when reaching the screen that holds the input field/s that the Auto-fill is intended to fill. Just press the button to run the Auto-fill.



2. From the main toolbar



Auto-fills can be run from the 'Play Auto-fill' button in the main toolbar. If you click on this button, a list of all available Auto-fills for the current screen will be unfolded.

Related Topics

[Creating an Auto-fill](#)
[Auto-fills Settings](#)

3.2.2 Auto-suggestion

Auto-suggestion is another useful and time-saving z/Scope feature. It consists in contextual menus that are displayed when typing into input fields. These pop-up menus contain a list of the recently typed phrases that match the user's initial keystrokes.

This feature resembles the functionality normally found when typing URLs into an internet browser address bar, and in numerous websites containing forms that require user input.

Once you start taking advantage of the Auto-Suggestion feature, you will be delighted at how easy terminal emulation is made with z/Scope.

To find out more about how this feature works, check the following sections:

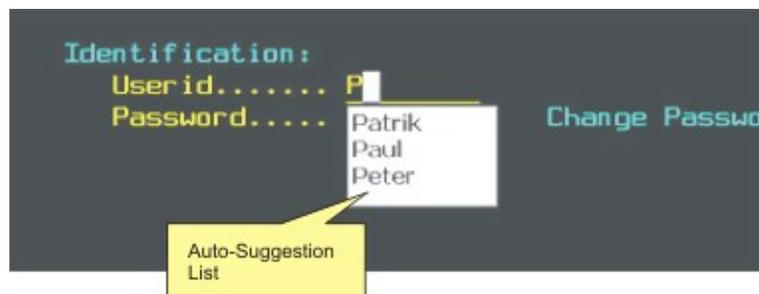
- [Using Auto-suggestion](#)
- [Deleting Auto-suggestion](#)

3.2.2.1 Using Auto-suggestion

First of all, in order to take advantage of the Auto-suggestion feature, you need to enable it by using the 'Turns Auto-Suggestion On/Off' button in the main toolbar. By default, connections are created with the Auto-Suggestion feature turned on.



Once the Auto-suggestion feature is enabled, it will store the words you type in the input fields of the emulation display. Next time you start typing into an input field, a list of suggestions of all possible variants of the string you are beginning to type will be displayed. Just scroll to the most suitable one and hit the 'Enter' key or double-click on it.



In the above screenshot you can see how the Auto-suggestion feature saves the user from having to re-type any previously entered information and thus you may get a glimpse of how this novel feature makes the terminal emulation experience more enjoyable.

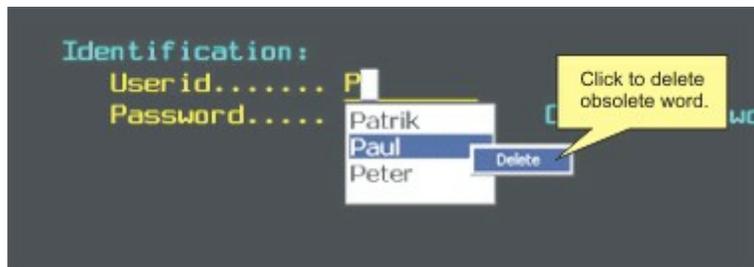
Related Topics

- Learn how to disable this feature for a particular connection in the [Preferences Tab in the Connection Settings dialog](#).

3.2.2.2 Deleting Auto-suggestions

There will be times when you may want to purge the Auto-suggestion list by deleting words that have become obsolete and you will not be using again, at least for a while.

If that is the case, all you have to do is right-click on the obsolete word in the Auto-suggestion list and select the 'Delete' option.



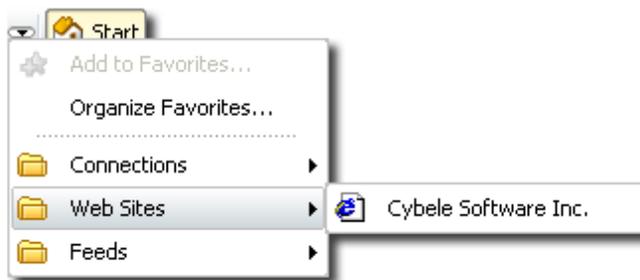
The deleted word will not appear again in the suggestion list until you type it again in an input field at a later time.

3.2.3 Favorites

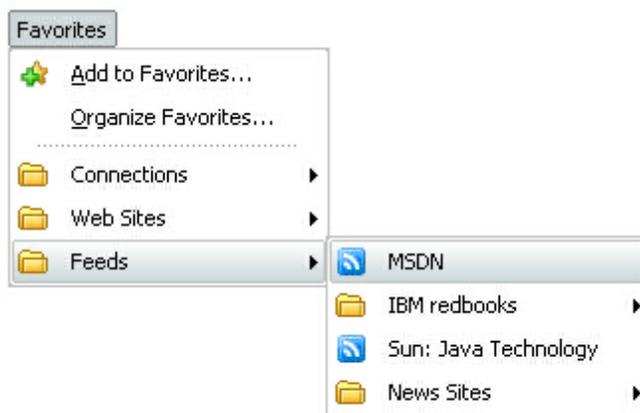
The favorites feature allows you to organize and easily access your most used connections, web sites, and RSS feeds.



You can access your Favorites from the main menu and the [Tab Selection toolbar](#):



You can also access your Favorites list from the main menu:



To learn how to create and configure your Favorites List, see the following topics:

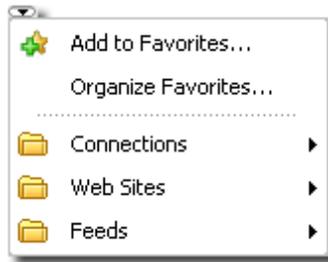
- [Building your Favorites list](#)
- [Organizing your Favorites list.](#)

3.2.3.1 Building Your Favorites List

There are four ways in which you can add connections, web sites and feeds to your Favorites list:

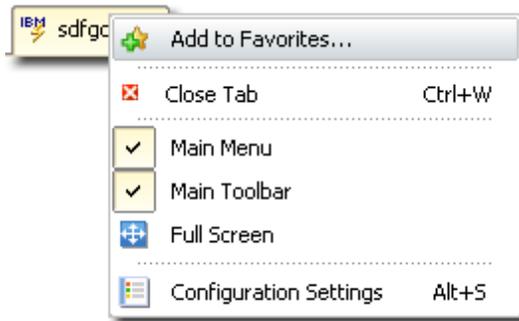
1. Through the toolbar button

With the connection, feed or web site as the active tab on the screen, press the arrow on the toolbar and then choose the 'Add to Favorites' option.



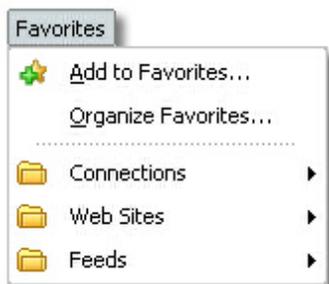
2. Through the tab's context popup menu

Right-click on the active tab for a connection, feed or web site that you want to add to your Favorite list and choose the 'Add to Favorites' option from the context popup menu.



3. Through the Favorites menu

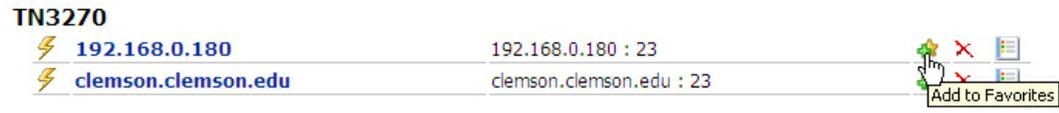
With the connection, feed or web site as the active tab on the screen, on the 'Favorites' menu choose the 'Add to Favorites' option:



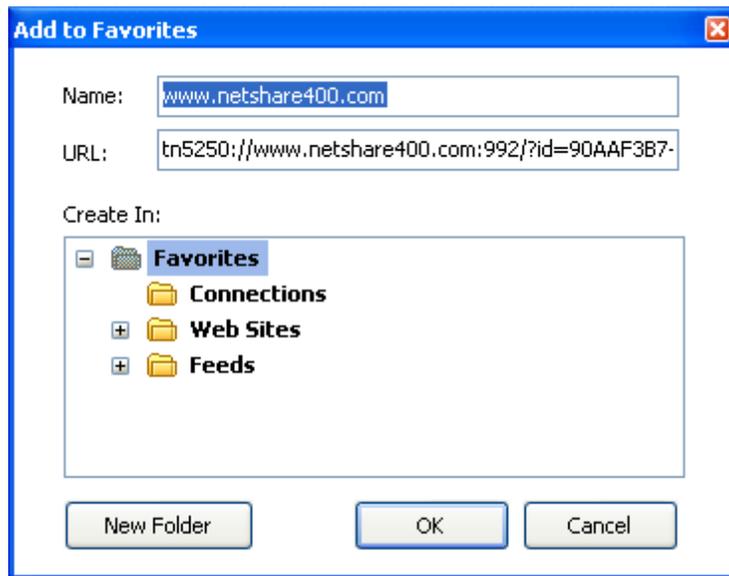
4. From the start page

This method is only useful to add connections to the Favorites list. From the start page, choose the connection you want to add, and click on the 'Add to Favorites' icon shown

below:



All of these options will open the 'Add to Favorites' dialog box, with the 'Name' and 'URL' parameters auto-completed with the active connection, feed or web site's information.



Name

This is a descriptive name for you to identify the connection. By default this will be completed with the name you have given to the connection or with the web site or feed's own title.

URL

The URL holds the connection path to the connection, feed or web site. By default this will be completed by the URL that belongs to the connection, feed or web site you invoked this dialog from.

Create in

Choose the folder in which you will like to place this connection, feed or web site.

New folder

Create new folders to better organize your Favorites list.

OK

Press 'OK' to confirm that you want to add this item to the Favorites list.

Cancel

Press 'Cancel' to discard the whole procedure.

Related Topics:

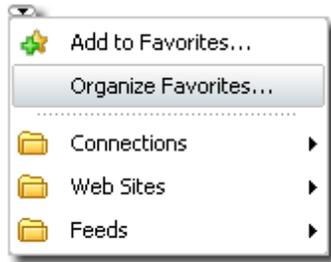
- To see how to access your Favorites, review the [Favorites Main Section](#).
- To learn how to organize your Favorites list, read onto [Organizing Your Favorites List](#).

3.2.3.2 Organizing Your Favorites List

There are two ways in which you can access the 'Organize your Favorites' dialog:

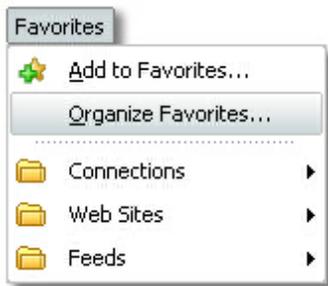
1. Through the toolbar button

Press the star button on the toolbar and then choose the 'Add to Favorites' option.

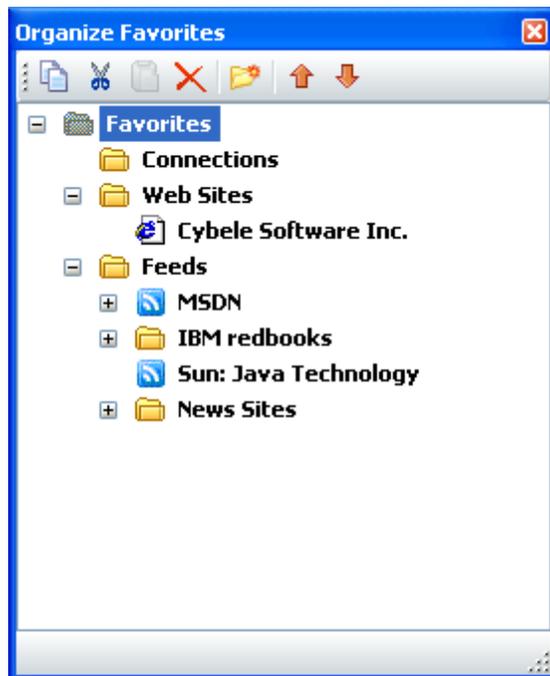


2. Through the Favorites menu

In the 'Favorites' menu choose the 'Organize Favorites' option:



These options will lead you to the 'Organize Favorites' dialog:



Drag and drop the Favorites and folders to move them to the desired folder. Use the arrows to expand or collapse the folder's contents.



Copy

Use this button to copy a connection, web site or feed to the clipboard.



Cut

Use this button to copy a connection, web site or feed to the clipboard and at the same time delete it from its current folder.



Paste

Use this button to paste a connection, web site or feed from the clipboard in the current folder.



Delete

Use this button to delete a connection, web site or feed from the Favorites list.



New Folder

Use this button to create a new folder in the Favorites list.



Move Up

Use this button to move a web site, connection or feed up inside the current folder.



Move Down

Use this button to move a web site, connection or feed down inside the current folder.

Related Topics:

- To see how to access your Favorites, review the [Favorites Main Section](#).
- To learn how to build your Favorites list, read onto [Building your Favorites List](#).

3.2.4 File Transfer

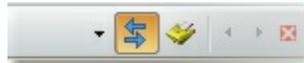
File transfer in z/Scope can be achieved mainly by two different methods. You can load the File Transfer Manager from the Main Toolbar of the Telnet Workspace; or if you need FTP capabilities, you can take advantage of the FTP Workspace that has been specifically equipped for that kind of tasks.

- File Transfer Using the File Transfer Manager
- [File Transfer Using the FTP Workspace](#)

Copyright © 2002-2012, Cybele Software Inc. All rights reserved.

3.2.4.1 Using the File Transfer Manager

When you need to start a file transfer on z/Scope Workbench edition, all you have to do is click on the 'Show File Transfer Manager' button in the connections toolbar, on the right hand side.



If a connection is already opened, you will also have the 'Upload/Download' buttons in the main toolbar.



File Upload

Use this button to upload a file from your PC to a tn3270 or tn520 host.

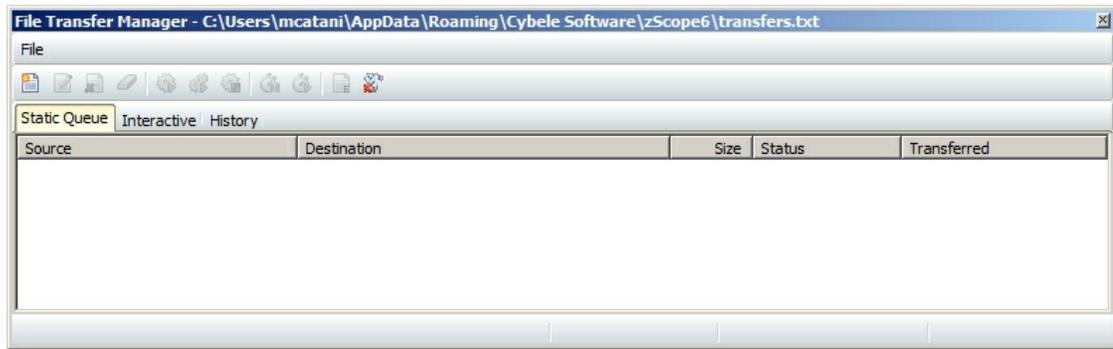


File Download

Use this button to download a file from a tn3270 or tn520 host to your PC.

The 'File Transfer' dialog will be shown, in which you can specify the file/s you want to Upload/Download, as well as many other settings.

This is how the File Transfer dialog looks like:



The File Transfer Manager allows you to transfer files using the following protocols: FTP, IND\$FILE, KERMIT, X-MODEM, Y-MODEM and Z-MODEM. It allows you to define the parameters for a particular file transfer job and then integrate that job into a Static Queue that contains all your previously defined file transfers jobs. You can then easily mark the desired jobs from the Static Queue to the Interactive and start transferring them with a single click. The File Manager will automatically keep a History of all the transfers.

The parameters available will vary according to the protocol you choose:

- [FTP](#)
- [IND\\$FILE](#)
- [KERMIT](#)
- [X-MODEM](#)
- [Z-MODEM](#)
- [Y-MODEM](#)

- To add a new File Transfer Job to the Queue, follow these steps:

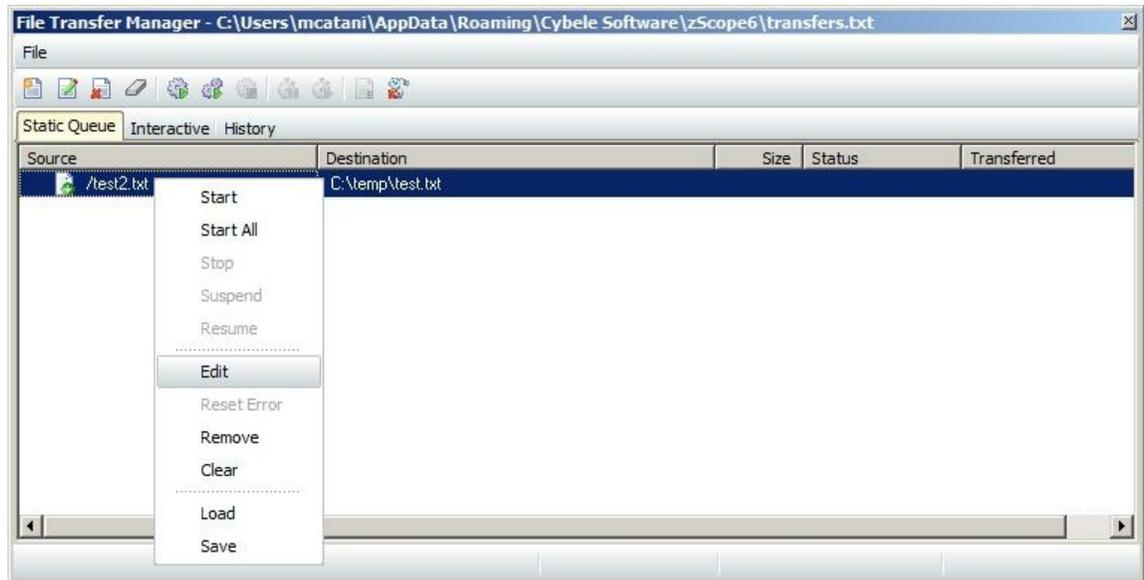
1. Click on the 'Add a New Transfer' button to launch the File Transfer Setup Wizard and click on Next.



2. Select the appropriate protocol ([FTP](#), [IND\\$FILE](#), [KERMIT](#), [X-MODEM](#), [Z-MODEM](#), [Y-MODEM](#)) and follow all the steps given.
3. Click on the 'Finish' button to save your file transfer settings.

The new file transfer job you specified will now appear in the Static Queue.

- To work with an already added Job, you can either use the buttons on the main menu or make a right click over it to see the available commands.



- The changes will be automatically applied to the queued item as you make them.
- To delete a file from the list, proceed this way:

1. Select the job you want to delete from the Queue.
2. Click on the 'Delete' button.

The job you selected will be removed from the Queue.

- To start transferring files, do the following:

1. Select the desired files in the Static Queue.
2. Click on the 'Start' button.

Those files that have been selected will be transferred. To transfer all files listed, click on the 'Start All' button.

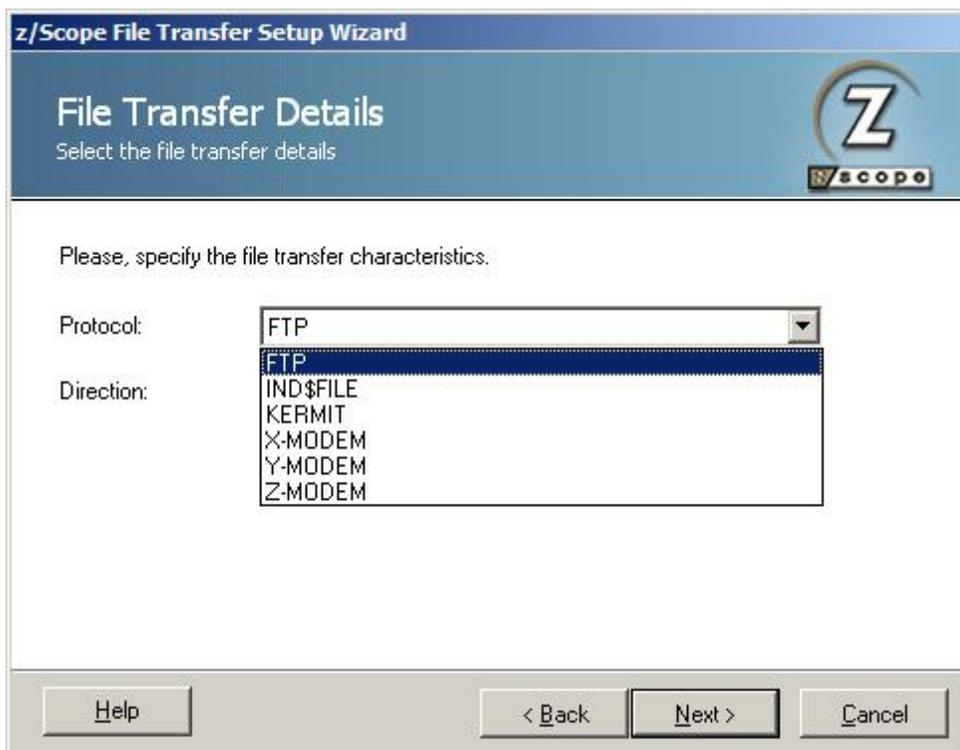
3.2.4.1.1 FTP

To add a new FTP File Transfer Job to the Queue, follow these steps:

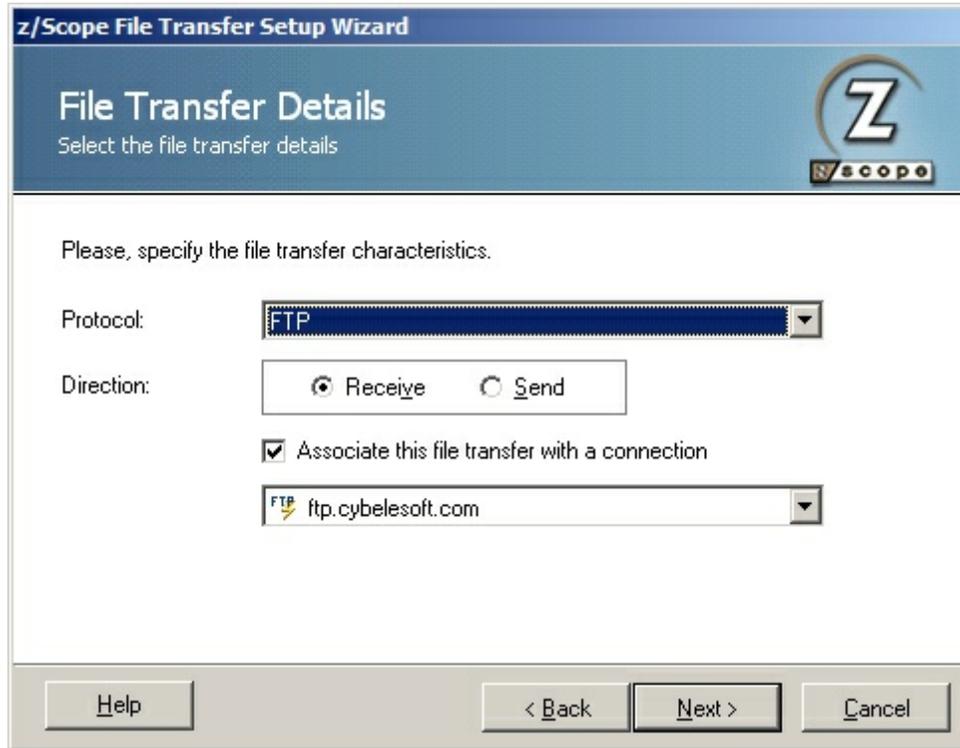
1. Click on the 'Add a New Transfer' button to launch the File Transfer Setup Wizard. Then click 'Next'.



2. Specify the file transfer characteristics. Set the protocol to FTP. Then click 'Next'.



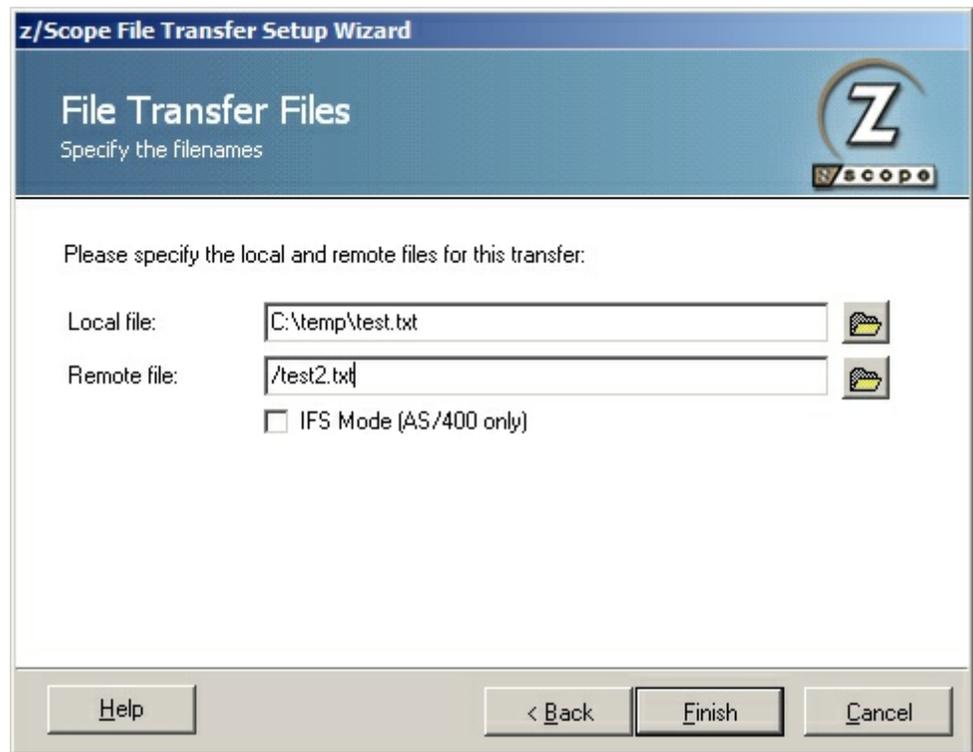
3. The Direction indicates if you are setting a download (RECEIVE) from the host to the PC, or an upload (SEND) from the PC to the host. Then you can select the option to associate the file transfer with an existing connection.



The image shows a screenshot of the 'z/Scope File Transfer Setup Wizard' dialog box. The title bar reads 'z/Scope File Transfer Setup Wizard'. The main window has a blue header with the text 'File Transfer Details' and 'Select the file transfer details'. The z/SCOPE logo is in the top right corner. Below the header, the text 'Please, specify the file transfer characteristics.' is displayed. The 'Protocol' is set to 'FTP' in a dropdown menu. The 'Direction' is set to 'Receive' with a selected radio button, and 'Send' is unselected. A checkbox labeled 'Associate this file transfer with a connection' is checked. Below this, a dropdown menu shows 'FTP ftp.cybelesoft.com'. At the bottom, there are four buttons: 'Help', '< Back', 'Next >', and 'Cancel'.

First Scenario:

If you associate the FTP File Transfer Job with an existing connection and click 'Next', the Job created will use all the settings predetermined on the connection you selected. As you do not need to enter any connectivity info, on the following screen you just need to specify the local and remote files and click 'Finish'.



Local File

In this field you must type a name for the file stored at the remote host.

Remote File

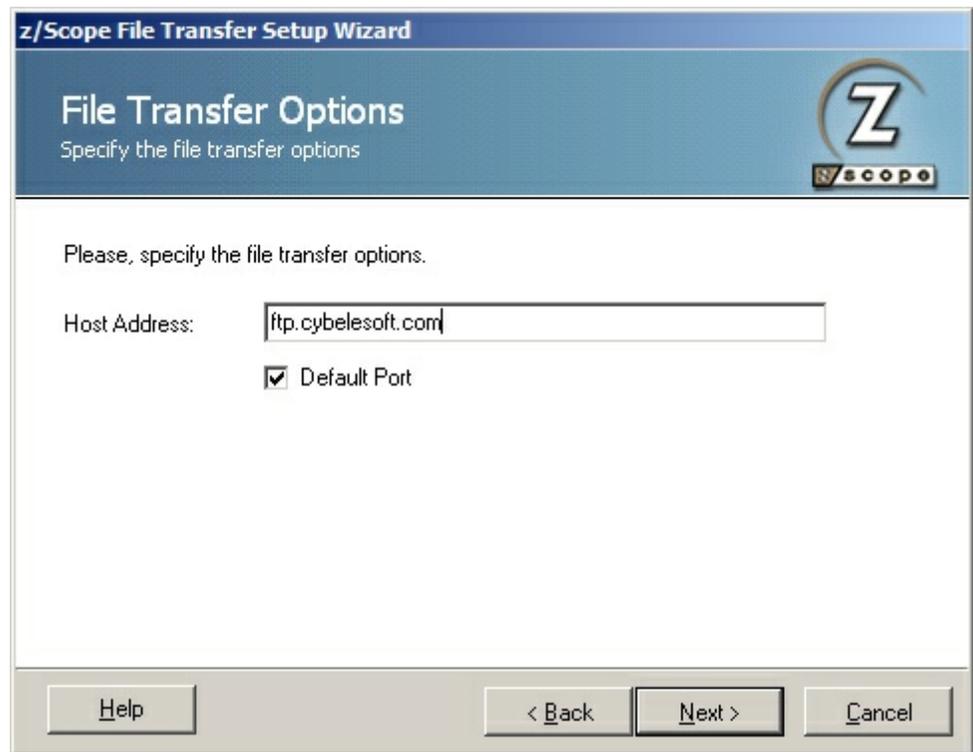
In this field you must type a name for the file stored on the local PC.

IFS Mode

This parameter works only on AS/400.

Second Scenario:

If you decide not to associate the file transfer with an existing connection, on the next step you must specify the Host and the Port.



Host

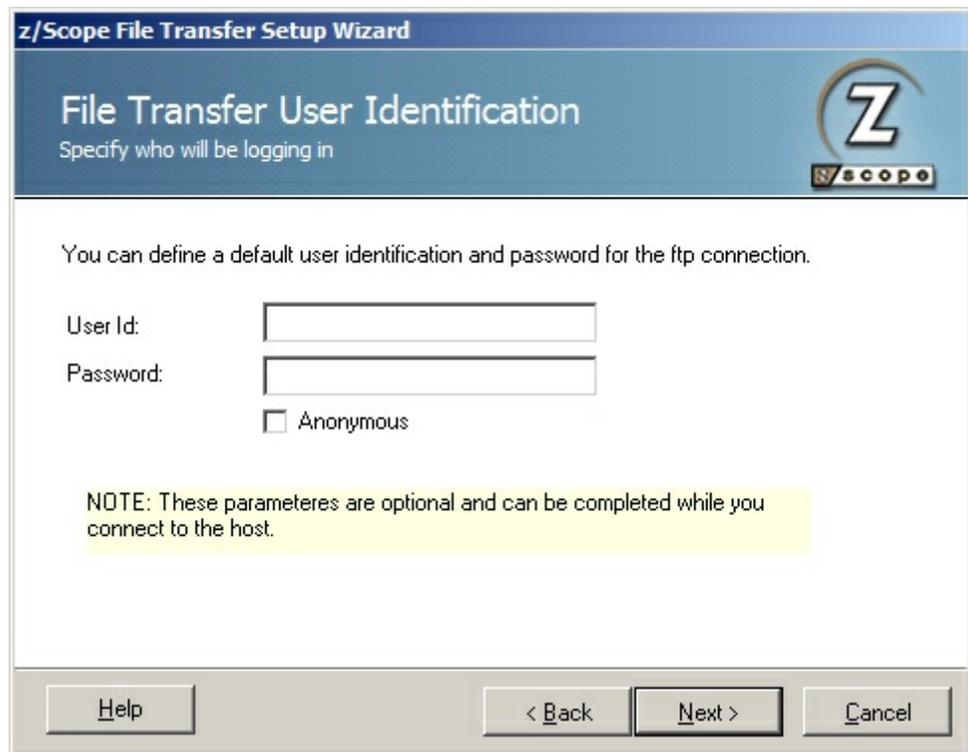
URL or IP address of the host machine that will act as the FTP server.

Port

TCP port defined for FTP access at the remote computer.

 Default FTP port number is 21.

- Once you defined the file transfer options, you must enter the User Identification.



The image shows a screenshot of the 'z/Scope File Transfer Setup Wizard' window. The title bar reads 'z/Scope File Transfer Setup Wizard'. The main heading is 'File Transfer User Identification' with the subtitle 'Specify who will be logging in'. The z/Scope logo is in the top right corner. The main text says: 'You can define a default user identification and password for the ftp connection.' Below this are three input fields: 'User Id:' with a text box, 'Password:' with a text box, and an unchecked checkbox labeled 'Anonymous'. A yellow highlighted note box contains the text: 'NOTE: These parameteres are optional and can be completed while you connect to the host.' At the bottom are four buttons: 'Help', '< Back', 'Next >', and 'Cancel'.

UserID

In this field you must enter your User ID.

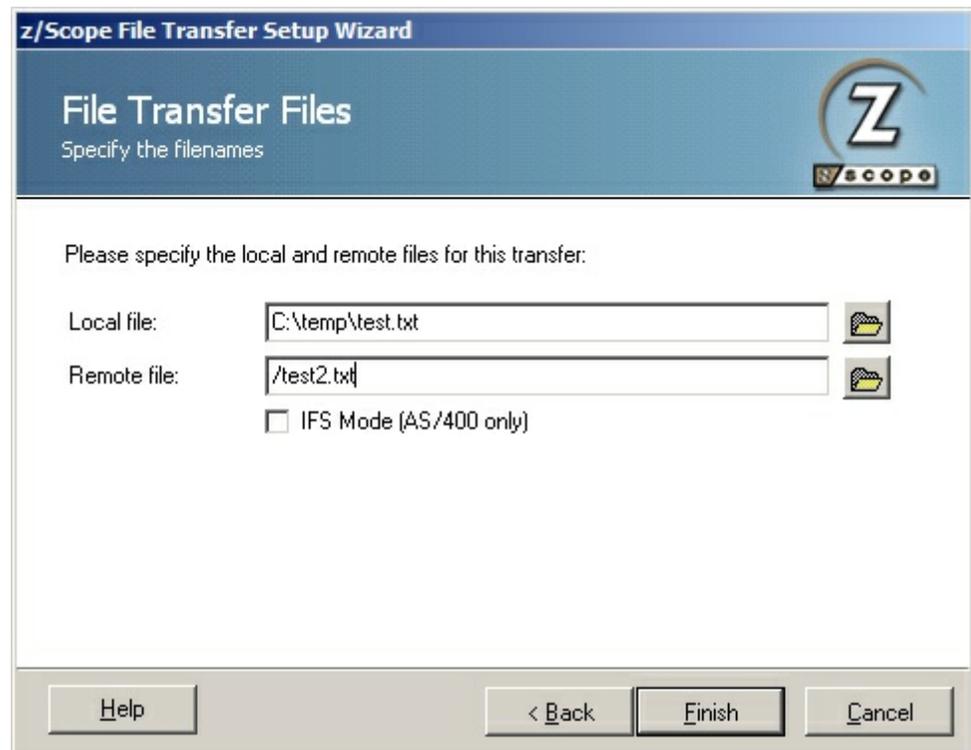
Password

In this field you must enter your Password.

Anonymous

Check this option if you prefer to log in anonymously.

- After that you will be taken to the last screen where you need to specify the local and remote files and click 'Finish'.



Local File

In this field you must type a name for the file stored at the remote host.

Remote File

In this field you must type a name for the file stored on the local PC.

IFS Mode

This parameter works only on AS/400.

Once you finish the Setup Wizard the new file transfer job you specified will appear in the Static Queue.

In the future, whenever you need to change the File Transfer settings you can either click the 'Edit' button on the main menu or make a right click on the File to select the 'Edit' option.

Inside the Edition Screen, besides all the parameters we explained above you will be able to change the appropriate transfer mode (ASCII/Binary/Auto). By default, all FTP connections will be set to 'Auto'.

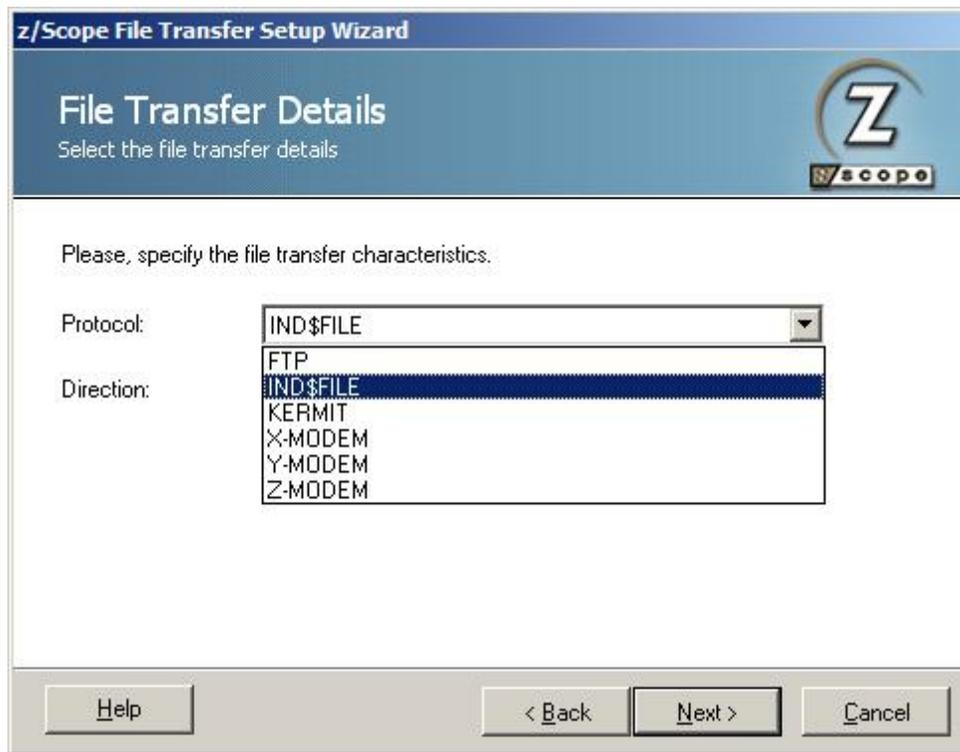
3.2.4.1.2 IND\$FILE

To add a new IND\$FILE File Transfer Job to the Queue, follow these steps:

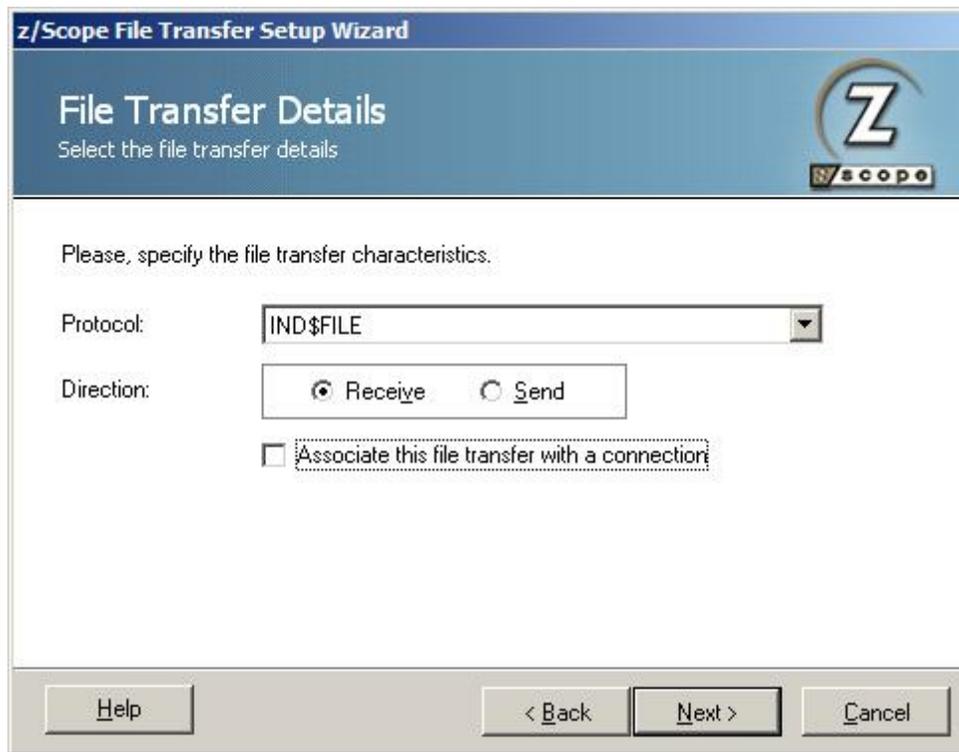
1. Click on the 'Add a New Transfer' button to launch the File Transfer Setup Wizard. Then click 'Next'.



2. Specify the file transfer characteristics. Set the protocol to IND\$FILE. Then click 'Next'.

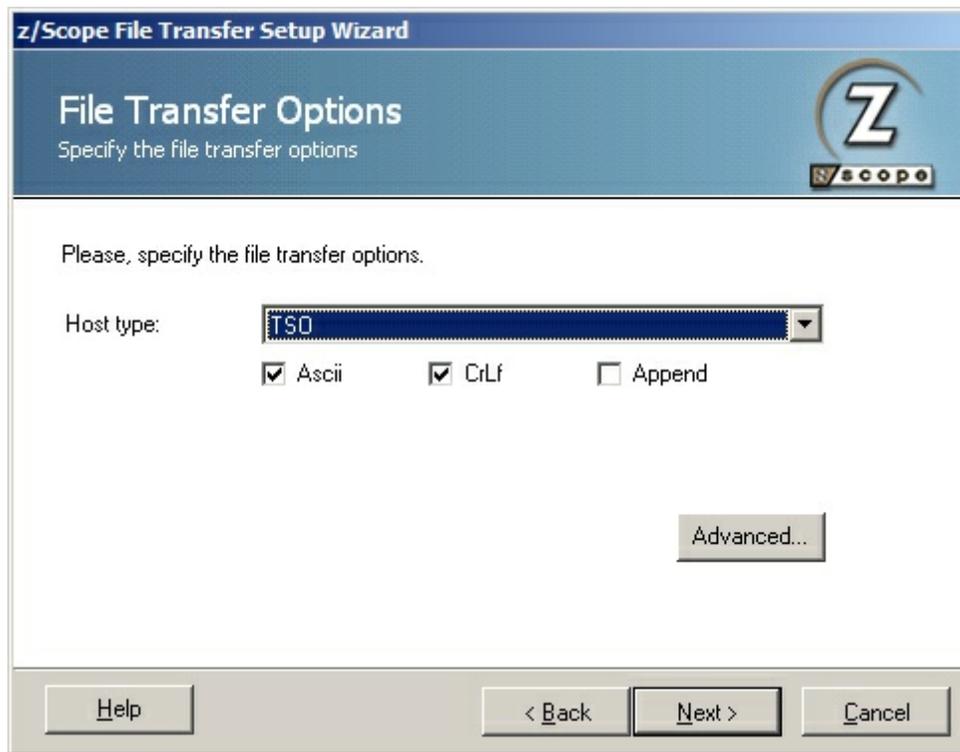


3. The Direction indicates if you are setting a download (RECEIVE) from the host to the PC, or an upload (SEND) from the PC to the host. You have the option to associate the file transfer transfer with an existing connection. You can either select it or leave it blank according to your preferences. Then click 'Next'.



The image shows a screenshot of the 'z/Scope File Transfer Setup Wizard' window. The title bar reads 'z/Scope File Transfer Setup Wizard'. The main heading is 'File Transfer Details' with the subtitle 'Select the file transfer details'. The z/Scope logo is in the top right corner. The main area contains the instruction 'Please, specify the file transfer characteristics.' followed by three controls: a 'Protocol' dropdown menu set to 'IND\$FILE', a 'Direction' section with 'Receive' selected (radio button) and 'Send' unselected, and an unchecked checkbox labeled 'Associate this file transfer with a connection'. At the bottom, there are four buttons: 'Help', '< Back', 'Next >', and 'Cancel'.

If you associate the FTP File Transfer Job to an existing connection and click 'Next', the Job created will use all the settings predetermined on the connection you selected. As you do not need to enter any connectivity info, on the following screen you just need to specify the local and remote files and click 'Finish'.



Host Type

In the combobox next to the IND\$FILE radio button you must select the type of host you are transferring to/from: VM/CMS, TSO or CICS.

ASCII

Check this option to specify that the file stored on the local PC in ASCII form is to be converted to EBCDIC during transfer to the host, and converted from EBCDIC to ASCII during transfer to the PC (needed for all non-binary file transfers).

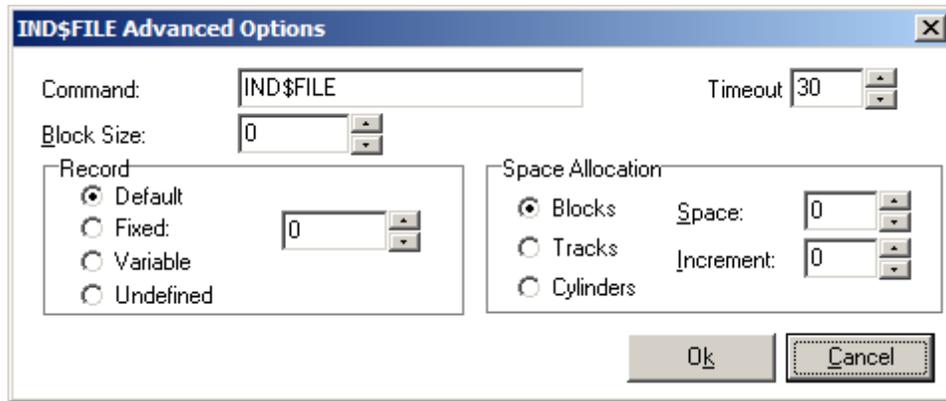
CRLF

Check this option to specify that carriage return/line feed should be recognized and deleted before file is stored on the host. It also deletes trailing spaces and inserts carriage return/line feed characters as the last two characters in a line when a file is stored on the PC.

Append

Allows you to append a PC file to the end of an OS data set, or an OS data set to the end of a PC file. This option is available only when transferring to/from TSO hosts.

- By clicking on the 'Advanced' tab you will find the following parameters:



Command

In this field you can change the name of the File Transfer program as it is defined in the host machine.

Timeout

Specify the amount of time (in seconds) that the program will attempt to connect.

Block Size

Specifies the block size of the TSO host data set. Enter the length of a data block in bytes.

Record

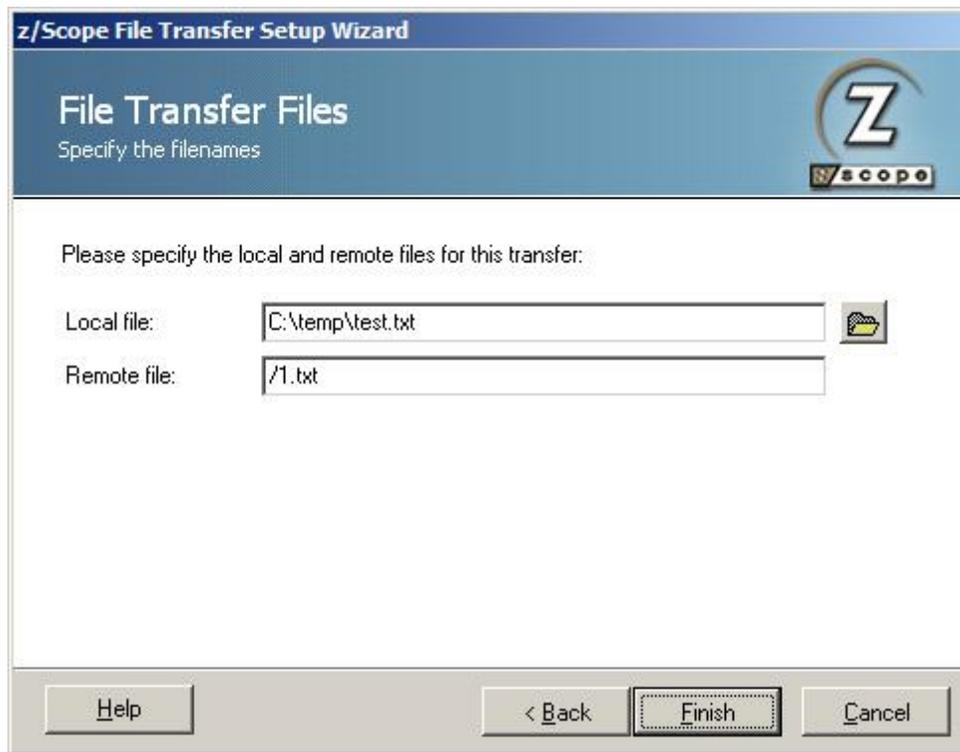
Specifies the record format for the data set. Available options are:

- Default-length records.
- Fixed-length records (you must enter the length manually).
- Variable-length records.
- Undefined-length records.

Space Allocation

Specifies an amount of space to be set aside for a new data set on a MVS/TSO volume. First, set the type of space you want (blocks, tracks, or cylinders) for the data set. Then provide the amount of space that you want the data set to occupy. If the data set needs more space than you asked for, provide the size of additional space (increment) to be used only when necessary.

- After that you will be taken to the last screen where you need to specify the local and remote files and click 'Finish'.



Local File

In this field you must type a name for the file stored at the remote host.

Remote File

In this field you must type a name for the file stored on the local PC.

IFS Mode

This parameter works only on AS/400.

Once you finish the Setup Wizard the new file transfer job you specified will appear in the Static Queue.

In the future, whenever you need to change the File Transfer settings you can either click the 'Edit' button on the main menu or make a right click on the File to select the 'Edit' option.

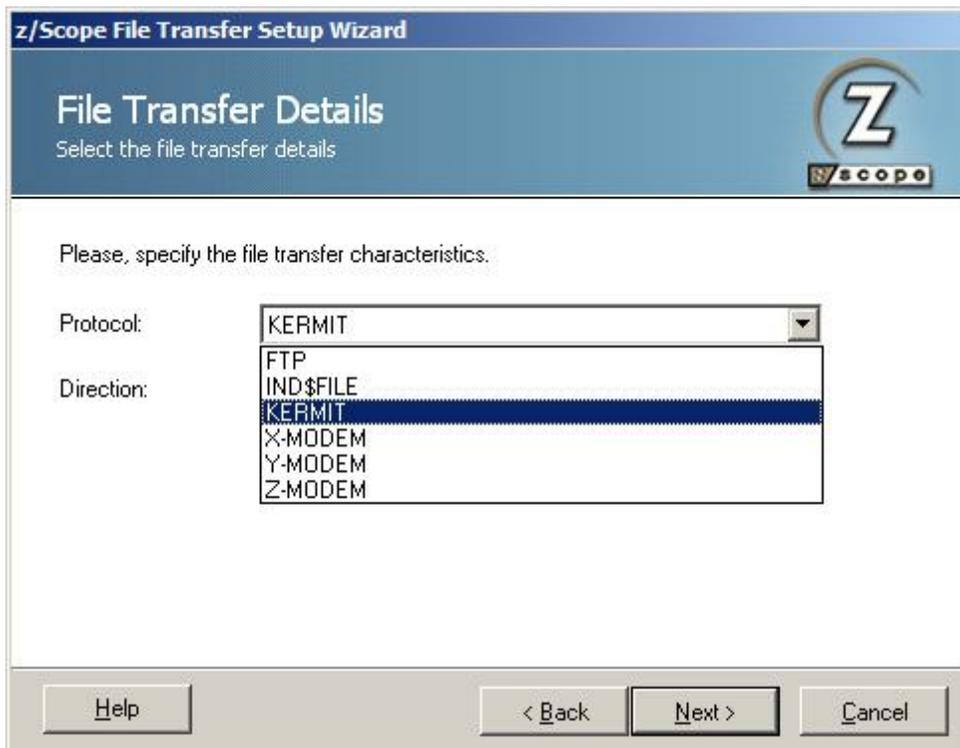
3.2.4.1.3 KERMIT

To add a new KERMIT File Transfer Job to the Queue, follow these steps:

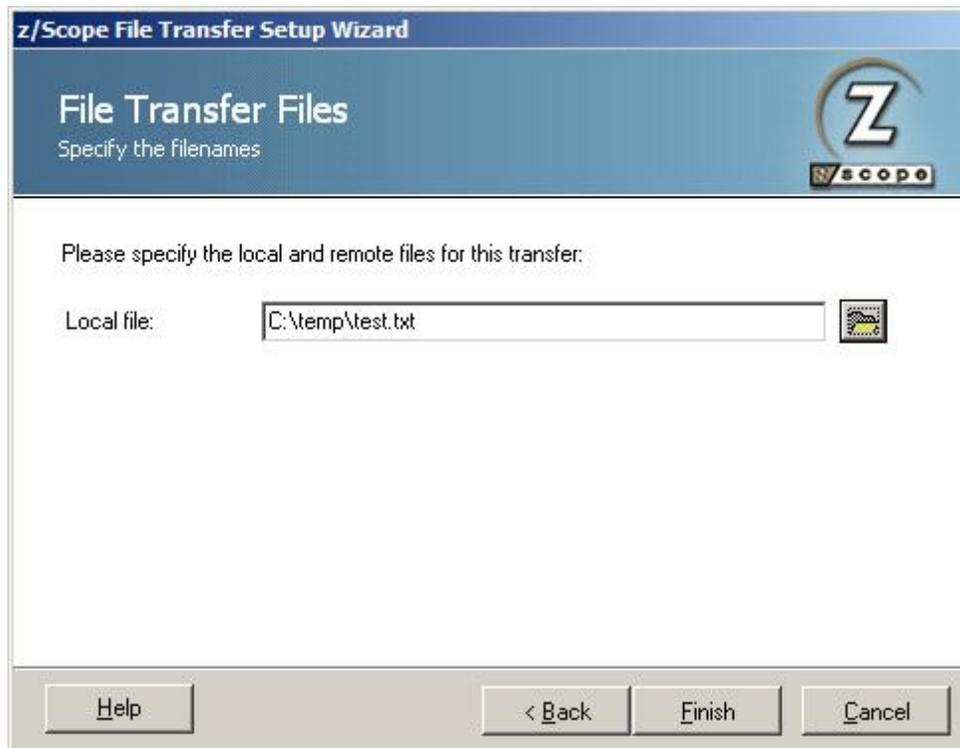
1. Click on the 'Add a New Transfer' button to launch the File Transfer Setup Wizard. Then click 'Next'.



2. Specify the file transfer characteristics. Set the protocol to KERMIT. Then click 'Next'.



3. The Direction indicates if you are setting a download (RECEIVE) from the host to the PC, or an upload (SEND) from the PC to the host. You have the option to associate the file transfer transfer with an existing connection. You can either select it or leave it blank according to your preferences. Then click 'Next'.
4. On the following screen you just need to specify the local file and click 'Finish'.



Once you finish the Setup Wizard the new file transfer job you specified will appear in the Static Queue.

In the future, whenever you need to change the File Transfer settings you can either click the 'Edit' button on the main menu or make a right click on the File to select the 'Edit' option.

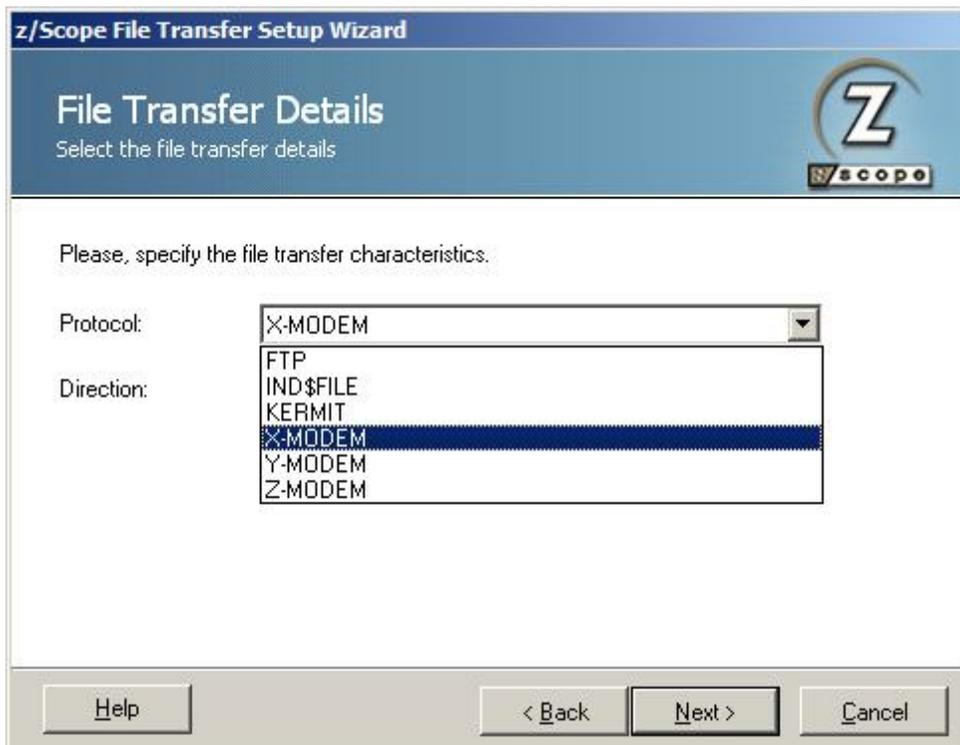
3.2.4.1.4 X-MODEM

To add a new X-Modem File Transfer Job to the Queue, follow these steps:

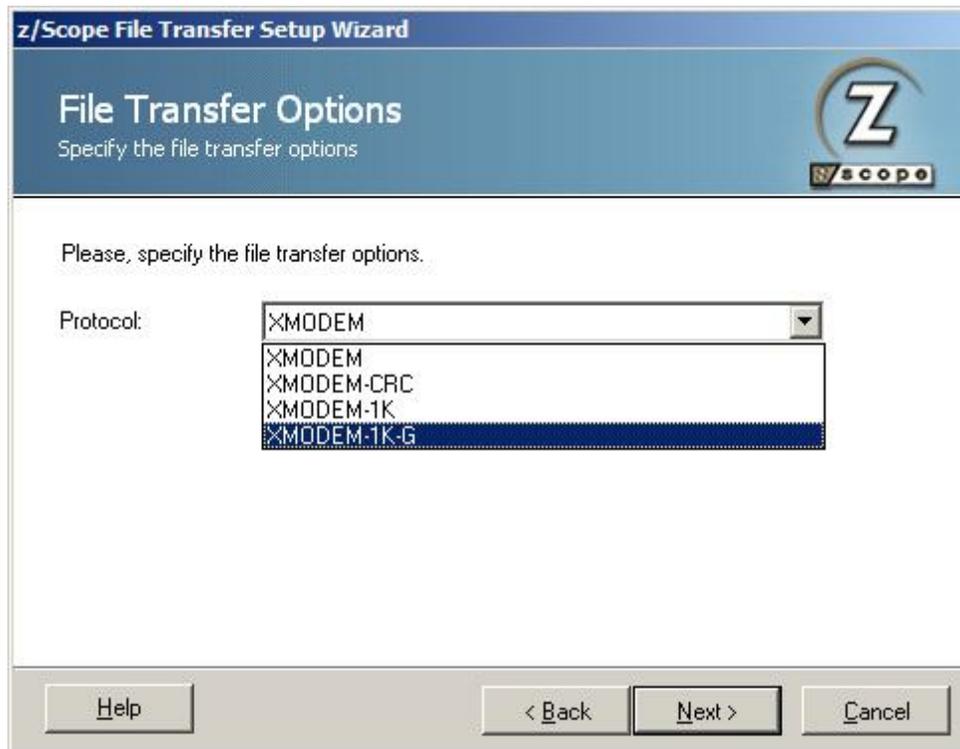
1. Click on the 'Add a New Transfer' button to launch the File Transfer Setup Wizard. Then click 'Next'.



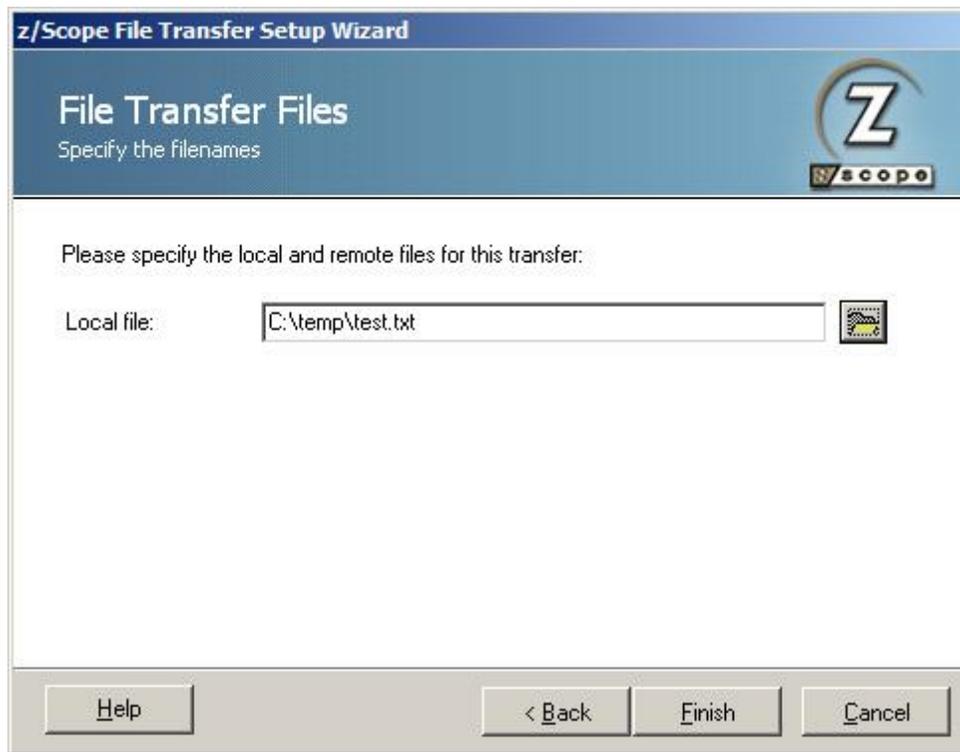
2. Specify the file transfer characteristics. Set the protocol to X-MODEM. Then click 'Next'.



3. The Direction indicates if you are setting a download (RECEIVE) from the host to the PC, or an upload (SEND) from the PC to the host. You have the option to associate the file transfer with an existing connection. You can either select it or leave it blank according to your preferences. Then click 'Next'.
4. Choose from the combobox the specific X-MODEM protocol you need to use: XMODEM, XMODEM-CRC, XMODEM-1K, XMODEM-1K-G. Then click 'Next'.



5. On the following screen you just need to specify the local file and click 'Finish'.



Once you finish the Setup Wizard the new file transfer job you specified will appear in the Static Queue.

In the future, whenever you need to change the File Transfer settings you can either click the 'Edit' button on the main menu or make a right click on the File to select the 'Edit' option.

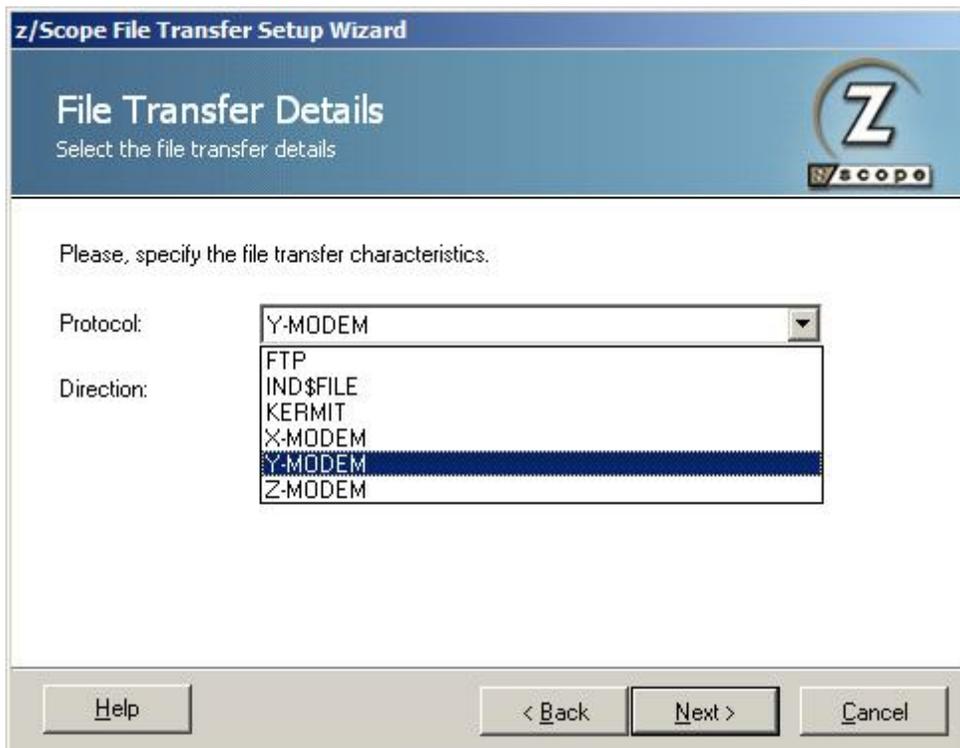
3.2.4.1.5 Y-MODEM

To add a new Y-Modem File Transfer Job to the Queue, follow these steps:

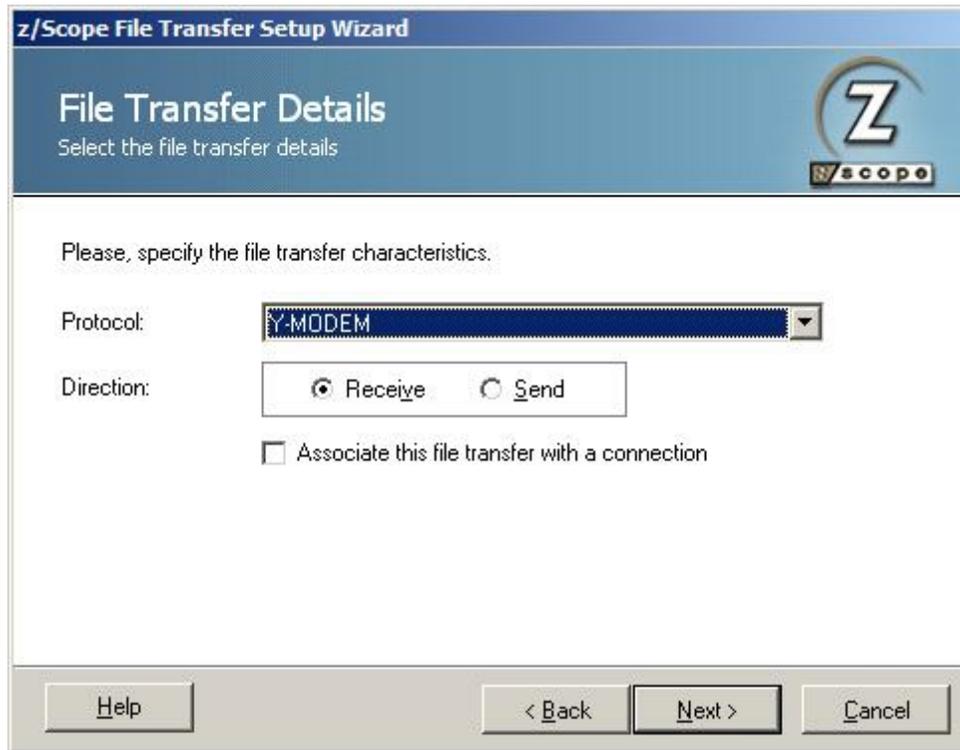
1. Click on the 'Add a New Transfer' button to launch the File Transfer Setup Wizard. Then click 'Next'.



2. Specify the file transfer characteristics. Set the protocol to X-MODEM. Then click 'Next'.

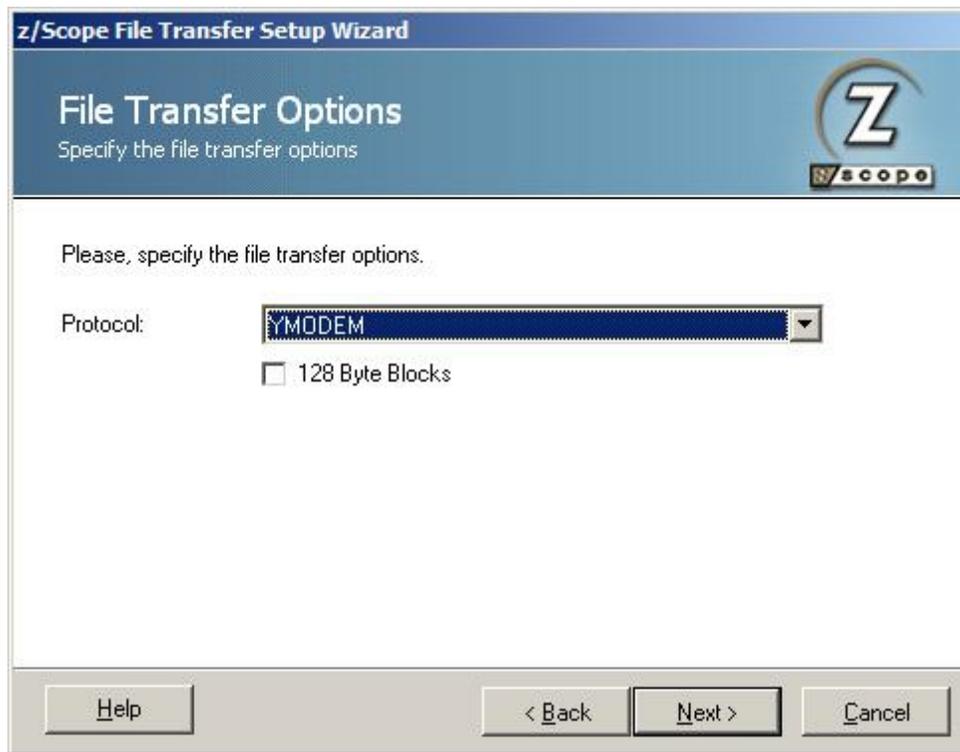


3. The Direction indicates if you are setting a download (RECEIVE) from the host to the PC, or an upload (SEND) from the PC to the host. You have the option to associate the file transfer with an existing connection. You can either select it or leave it blank according to your preferences. Then click 'Next'.



The image shows a screenshot of the 'z/Scope File Transfer Setup Wizard' dialog box, specifically the 'File Transfer Details' step. The title bar reads 'z/Scope File Transfer Setup Wizard'. The main title is 'File Transfer Details' with the subtitle 'Select the file transfer details'. The z/Scope logo is in the top right corner. The main area contains the instruction 'Please, specify the file transfer characteristics.' Below this, there are three settings: 'Protocol' is a dropdown menu currently showing 'Y-MODEM'; 'Direction' has two radio buttons, 'Receive' (which is selected) and 'Send'; and a checkbox labeled 'Associate this file transfer with a connection' which is currently unchecked. At the bottom, there are four buttons: 'Help', '< Back', 'Next >', and 'Cancel'.

4. Choose from the combobox the specific Y-MODEM protocol you need to use: YMODEM or YMODEM-G. There is also an option to make transfers using blocks of 128 bytes. Once you select the settings you need, click on 'Next'.



5. On the following screen you just need to specify the local file and click 'Finish'.



Once you finish the Setup Wizard the new file transfer job you specified will appear in the Static Queue.

In the future, whenever you need to change the File Transfer settings you can either click the 'Edit' button on the main menu or make a right click on the File to select the 'Edit' option.

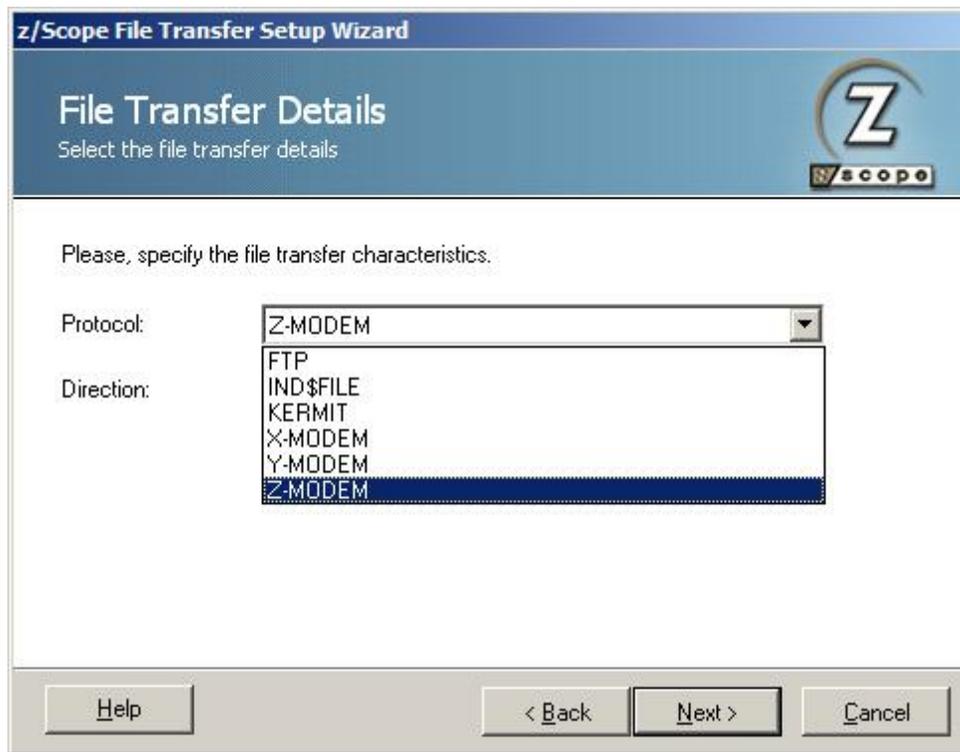
3.2.4.1.6 Z-MODEM

To add a new X-Modem File Transfer Job to the Queue, follow these steps:

1. Click on the 'Add a New Transfer' button to launch the File Transfer Setup Wizard. Then click 'Next'.

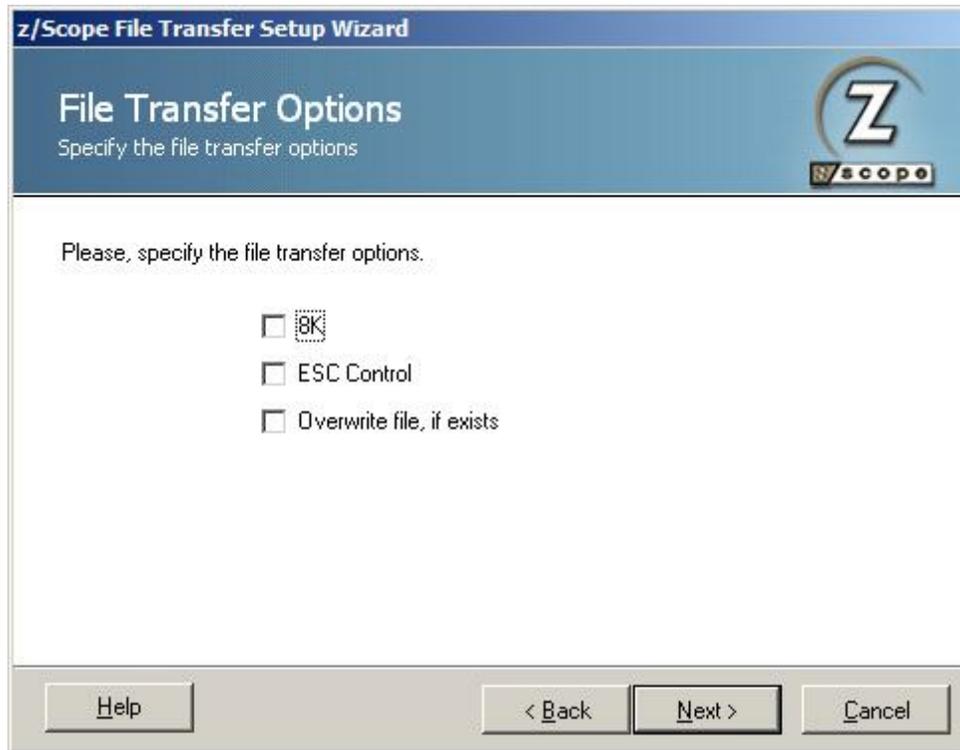


2. Specify the file transfer characteristics. Set the protocol to Z-MODEM. Then click 'Next'.



3. The Direction indicates if you are setting a download (RECEIVE) from the host to the PC, or an upload (SEND) from the PC to the host. You have the option to associate the file transfer with an existing connection. You can either select it or leave it blank according to your preferences. Then click 'Next'.

4. Specify the file transfer options. Then click 'Next'.



8K

Check this option to enable 8K blocks.

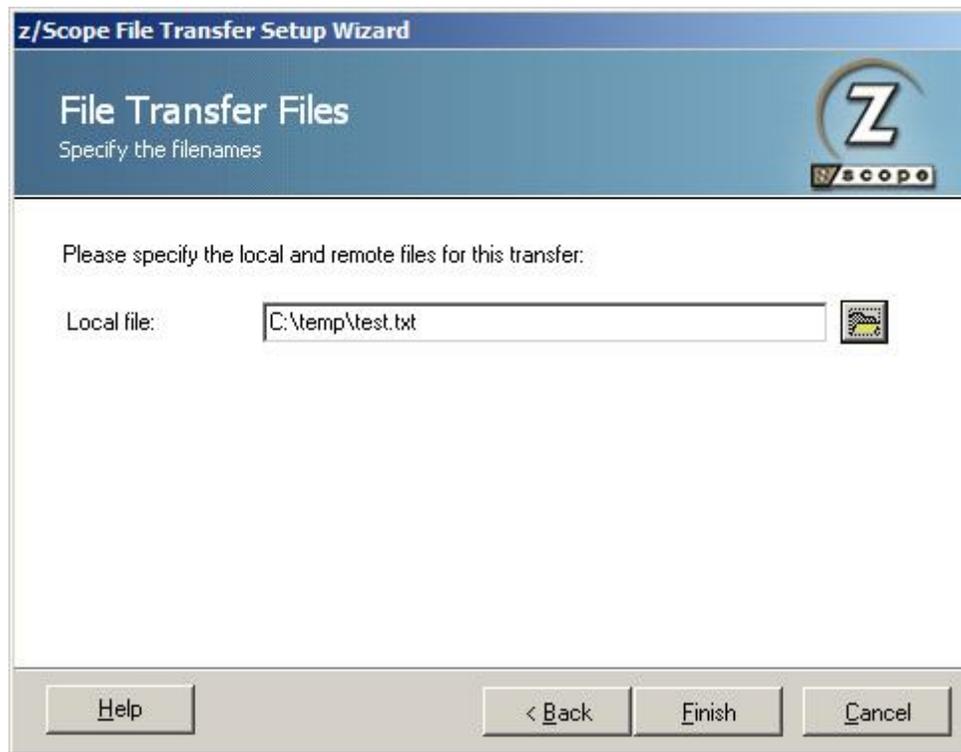
ESC Control

Check this option if you would like to **escape all control chars**. Unchecked, control chars will not be transformed but taken as part of the file.

Overwrite file, if exists.

Check this option if you would like the transferred file to overwrite an existing file in case they have the same name.

5. On the following screen you just need to specify the local file and click 'Finish'.



Once you finish the Setup Wizard the new file transfer job you specified will appear in the Static Queue.

In the future, whenever you need to change the File Transfer settings you can either click the 'Edit' button on the main menu or make a right click on the File to select the 'Edit' option.

3.2.4.2 Using the FTP Workspace

For a complete description of the elements of z/Scope's FTP workspace, you may want to check the following section:

- [The FTP User Interface](#)

To learn how to get started setting up FTP connections, check the following topics:

- [Setting up an FTP Connection](#)
- [Editing an FTP Connection](#)

In this topic, file transfer operations from within the FTP workspace will be discussed. There are 3 ways in which you can start a file transfer:

1. Using the FTP toolbar buttons



To start a file transfer using the FTP toolbar Upload/Download buttons, follow these steps:

- If you want to upload files from your local computer to the remote computer:
 1. Select the file/s you want to upload from the 'My computer' panel.
 2. Select the appropriate path in the 'FTP server' panel for the file/s to be saved.
 3. Click on the 'Upload' button in the FTP toolbar.

The file/s you selected will be placed in the queue and the transfer will be started automatically into the selected folder in the FTP server panel.

- If you want to download files from the remote computer to your local computer:
 1. Select the file/s you want to download from the 'FTP Server' panel.
 2. Select the appropriate path in the 'My computer' panel for the file/s to be saved.
 3. Click on the 'Download' button in the FTP toolbar.

The file/s you selected will be placed in the queue and the transfer will be started automatically into the selected folder in the My computer panel.

2. Using Drag & Drop between panels



To start a file transfer using the drag and drop method, follow these steps:

- To upload files from your local computer to the remote computer:
 1. Select the file/s you want to upload from the 'My computer' panel.
 2. Select the appropriate path in the 'FTP server' panel for the file/s to be saved.
 3. Drag and drop the file/s from the 'My computer' panel to the location selected in the previous step.

The file/s you selected will be placed in the queue and the transfer will be started automatically.



Optionally, you can drag and drop the file/s to the 'File Transfer Queue' instead of the 'My Computer' panel. In this way, the transfer will be queued until you choose to start it, as described in method #3 ('Starting a previously queued transfer').

- To download files from the remote computer to your local computer
 1. Select the file/s you want to download from the 'FTP Server' panel.
 2. Select the appropriate path in the 'My computer' panel for the file/s to be saved.

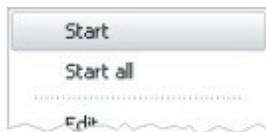
3. Drag and drop the file/s from the 'FTP Server' panel to the location selected in the previous step.

The file/s you selected will be placed in the queue and the transfer will be started automatically.



Optionally, you can drag and drop the file/s to the 'File Transfer Queue' instead of the 'My Computer' panel. In this way, the transfer will be queued until you choose to start it, as described in method #3 ('Starting a previously queued transfer').

3. Starting a previously queued transfer

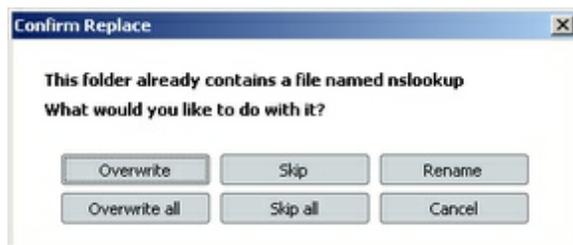


You can drag and drop any file to the 'File Transfer Queue' so that the transfer remains queued until you choose to start it.

- To start a previously queued transfer, do the following:
 1. Right-click on the file in the queue that you want to start transferring.
 2. Select the 'Start' option from the context menu.

The selected file will start transferring instantly.

When you attempt to transfer files that already exist in the destination folder, the following dialog will be displayed:



Overwrite

Select this option if you want to overwrite the destination file with the new one.

Overwrite all

Select this option if you are transferring many files at the same time and you don't want to be asked again for confirmation if a file with the same name already exists in the destination folder. Files in the destination folder will be overwritten.

Skip

Select this option if you are transferring many files at the same time and you don't want to overwrite the existing file with the new one.

Skip all

Select this option if you are transferring many files at the same time and you don't want any of the existing files to be overwritten by the new ones.

Rename

Select this option to rename the new file so that the existing one won't be overwritten.

Cancel

Select this option to discard the whole transfer process.

3.2.5 HotSpots

HotSpots are a very useful feature that allows the user to mouse-enable the emulation display by placing point-and-click controls that provide access to screen commands otherwise only accessible through keyboard commands.

z/Scope includes several pre-defined HotSpots, and at the same time provides an easy and intuitive interface to create as many new user-defined HotSpots as required.

To learn more about the HotSpots feature, read the next topic:

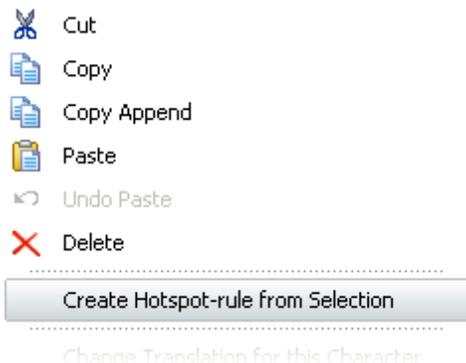
- [Creating a HotSpot](#)
- [Using HotSpots](#)

Related Topics

[HotSpots Settings](#)

3.2.5.1 Creating a HotSpot

HotSpots can be easily created by directly right-clicking any text string in the screen and selecting the 'Create HotSpot-rule from Selection' option from the context menu.



The 'HotSpots Wizard' will be launched and you will be presented with the Welcome

screen.



Click on 'Next' to go to the next step, in which you will have to set the following parameters:



Hot text

In this field you must type a rule in the form of a Regular Expression to match the string you want to turn into a Hotspot. If you are not familiar with Regular Expressions, see [Regular Expressions Appendix](#).

Case sensitive

Check this option if you want to consider the case when matching the Hot text.

Text is followed/preceded by a space

Check these options to specify that the Hot text will only be matched if preceded or followed by a blank character.

Click on 'Next' to go to the next step, in which you will have to choose the type of action that will be performed when the HotSpot is pressed.

- Send Keystrokes: Select this option if you want to specify keystrokes to be sent in response to the HotSpot activation.
- Start a Macro: Select this option if you want to specify a Macro to be started in response to the HotSpot activation. This allows you to run a macro with a simple mouse click

If the 'Send Keystrokes' option is selected you will have to fill in the following parameters:

The screenshot shows the 'z/Scope Hotspot Setup Wizard' dialog box, specifically the 'Rule Action' step. The title bar reads 'z/Scope Hotspot Setup Wizard'. The main heading is 'Rule Action' with the subtitle 'Specifying the hotspot-rule action'. The z/SCOPE logo is in the top right corner. The dialog contains the following elements:

- A label 'Specify the type of action:' followed by a dropdown menu currently set to 'Send Keystrokes'.
- A label 'Specify keys to type. Function keys must be enclosed in {}'.
- Two rows of input fields: 'Host Keystrokes:' and 'VT Keystrokes:'. Each row has a text input field containing 'ACCESS', a plus sign button, and a dropdown menu.
- At the bottom, there are three buttons: 'Help', '< Back', and 'Next >', and a 'Cancel' button on the far right.

Host Keystrokes

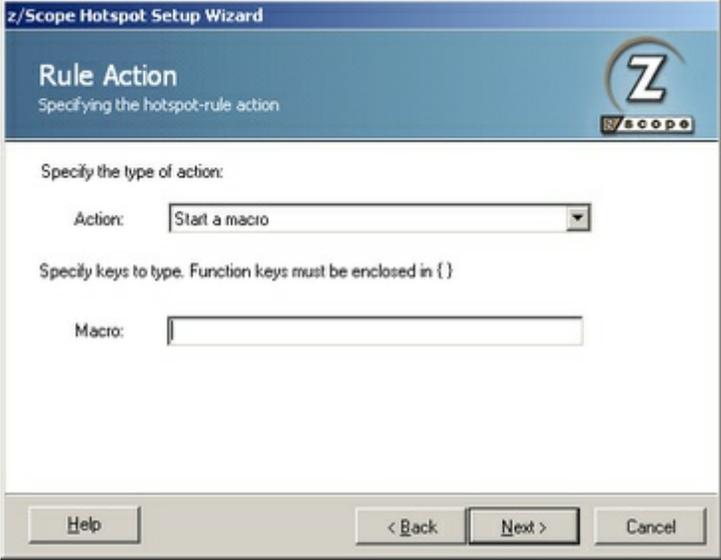
In this field you can type the text that you want to be sent as keystrokes for 3270 and 5250 hosts.

VT Keystrokes

In this field you can type the text that you want to be sent as keystrokes for VT hosts.

! To add a function to any of these fields, select it from the key list and press the '+' button next to it. Functions can also be entered manually by enclosing them in brackets (ie. {ENTER}).

If the 'Start a macro' option is selected, the dialog will look like this:

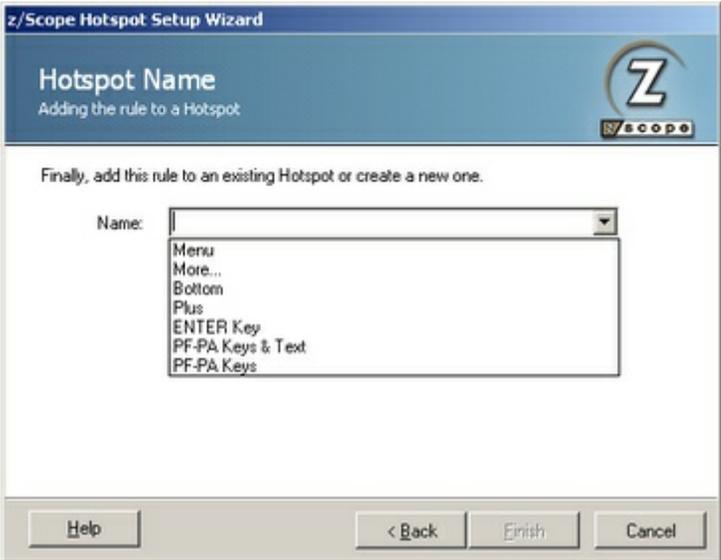


The screenshot shows the 'z/Scope Hotspot Setup Wizard' dialog box, specifically the 'Rule Action' step. The title bar reads 'z/Scope Hotspot Setup Wizard'. The main header area contains the text 'Rule Action' and 'Specifying the hotspot-rule action', along with the z/Scope logo. The main content area has the instruction 'Specify the type of action:' followed by a dropdown menu labeled 'Action:' with 'Start a macro' selected. Below this is the instruction 'Specify keys to type. Function keys must be enclosed in {}' followed by a text input field labeled 'Macro:'. At the bottom, there are four buttons: 'Help', '< Back', 'Next >', and 'Cancel'.

Macro

In this field you can type the name of a Macro to be started in response to the HotSpot activation.

Click on 'Next' to go to the last step of the HotSpot creation process.



The screenshot shows the 'z/Scope Hotspot Setup Wizard' dialog box, specifically the 'Hotspot Name' step. The title bar reads 'z/Scope Hotspot Setup Wizard'. The main header area contains the text 'Hotspot Name' and 'Adding the rule to a Hotspot', along with the z/Scope logo. The main content area has the instruction 'Finally, add this rule to an existing Hotspot or create a new one.' followed by a dropdown menu labeled 'Name:'. The dropdown menu is open, showing a list of options: 'Menu', 'More...', 'Bottom', 'Plus', 'ENTER Key', 'PF-PA Keys & Text', and 'PF-PA Keys'. At the bottom, there are four buttons: 'Help', '< Back', 'Finish', and 'Cancel'.

To create a new HotSpot from the defined rule, enter the name of the new HotSpot in the 'Name' field; otherwise select one of the existing Hotspots from the list to add the defined rule to it.

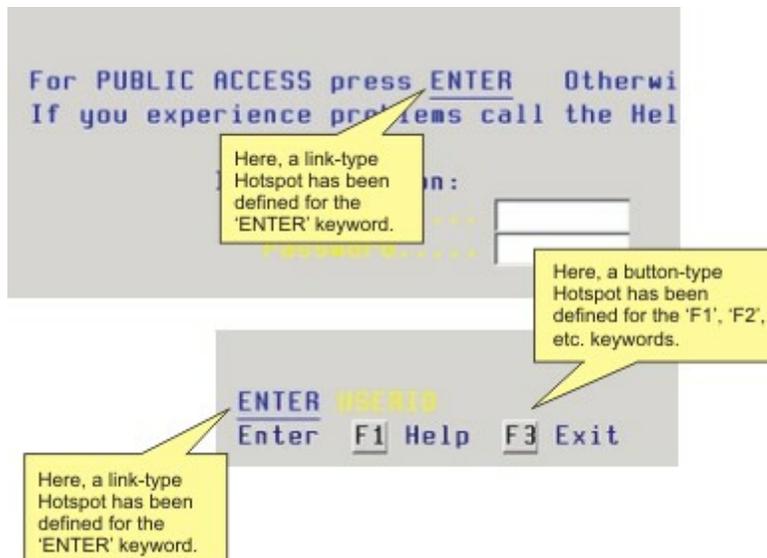
Click on 'Finish' to confirm the new HotSpot creation. Otherwise, click on 'Cancel' to discard the whole procedure.

Related Topics

- [Hotspots Settings](#)
- [Using HotSpots](#)

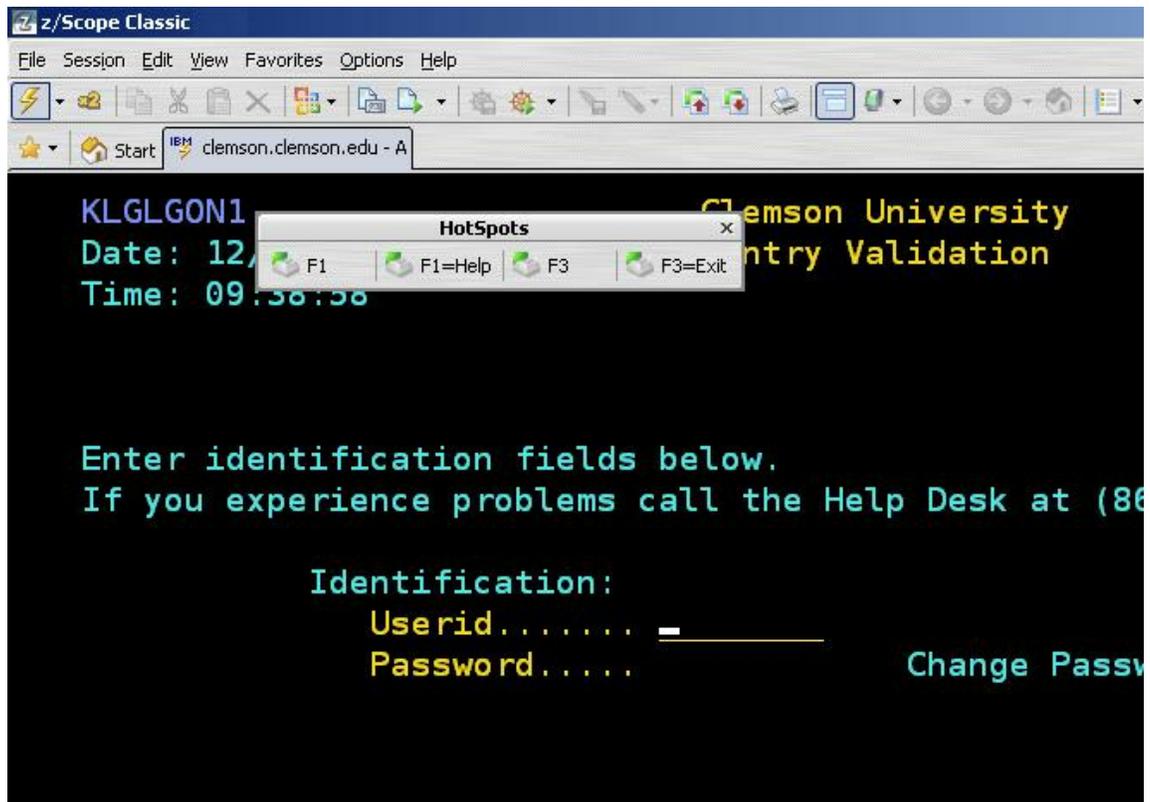
3.2.5.2 Using HotSpots

The HotSpots that you created for a particular screen will appear directly on that screen in an in-line manner, as you can see in the screenshots below.



The way HotSpots appear on the screen depends on the display format specified in the [Style tab in HotSpot Settings](#).

HotSpots can be configured to appear inside a dynamic toolbar at the top of the emulation display (as shown in the screenshot below) from the [Style tab in HotSpot Settings](#).



 In order to be able to use your HotSpot within a connection, you must first enable the HotSpot for that particular connection from the [Preferences Tab in Connections Settings \(3270, 5250\)](#) or [Preferences Tab in Connections Settings \(vt\)](#) dialog.

Related Topics

- [Hotspots Settings](#)
- [Creating a HotSpot](#)

3.2.6 Keyboard Support

An important aspect of z/Scope is its compatibility and support for a wide range of standard keyboards such as IBM, EXTRA, IRMA and RUMBA.

z/Scope gives you full control and customization of the keyboard by allowing you to re-map virtually any key to new combination of different keys and even mouse clicks!

It also allows you to define keyboard shortcuts to z/Scope's main features such as Macros and Navigators.

Related Topics

- [Keyboard Settings](#)

3.2.7 Keypads

Keypads are a useful tool that enables the user to substitute host commands with mouse clicks. Together with [HotSpots](#), Keypads integrate a set of exciting features designed to bring you the most convenient mouse-enabled terminal emulation capabilities.

To learn more about the Keypads feature, go to the next topic:

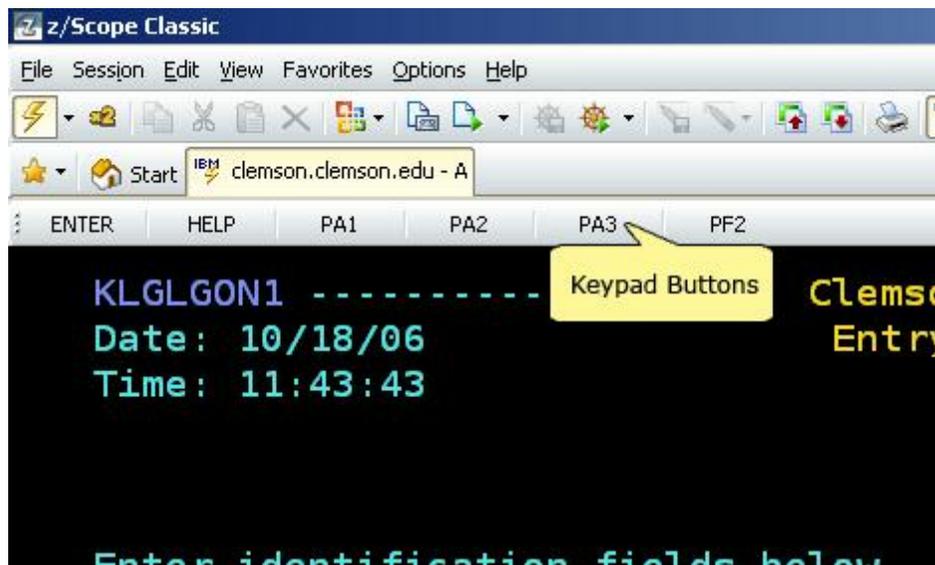
- [Using Keypads](#)

Related Topics

- [Keypads Settings](#)

3.2.7.1 Using Keypads

Keypads that are available for a particular connection will automatically appear in their corresponding toolbar at the top of the emulation display, as shown in the screenshot below.



From now on, you can click on the Keypad buttons instead of hitting the keys in your keyboard.

 In order to be able to use your Keypad within a Connection, you must first enable the Keypad for that particular connection in the [Keypads tab in the Connections Settings](#) dialog.

Related Topics

- [Keypads Settings](#)

3.2.8 Macros

Macros are sequences of keystrokes sent to the mainframe that are recorded so you can then reproduce them by a simple mouse click.

Like Navigators, macros are useful when you need to automate a sequence of commands so you don't have to re-type the same commands many times.

To learn more about Macros, read the following topics:

- [Creating a Macro](#)
- [Using a Macro](#)
- [Debugging a Macro](#)

Related Topics

[Macros Settings](#)

3.2.8.1 Creating a Macro

The Macro creation process is very simple. When you want to start recording a Macro, click on the 'Start Macro Recording' button in the main toolbar. From now on, every action will be recorded as part of the new macro.



Once pressed, this button will be replaced by the 'Stop Macro Recording' button.

After you have entered the last command you want to include in the macro, click on the 'Stop Macro Recording' button in the main toolbar.



Once pressed, this button will be replaced by the 'Start Macro Recording' button.

No more actions will be recorded, and you will be prompted for a name for the new macro.



Enter a name and click 'OK'. Or click 'Cancel' to discard the whole procedure.

 All macros saved for a particular connection will be listed in the arrow next to the 'Play Macro' button in the main toolbar. See [Using Macros](#).

To learn how to use the macros that you create, see the next topic:

- [Using Macros](#)

Related Topics

[Macros Settings](#)

3.2.8.2 Using Macros

In order to run one of the available macros for the current connection, select it from the scroll-down list of the 'Run Macro' button in the toolbar.



 Once pressed, this button will be replaced by the 'Stop Macro Playback' button.

The keystroke sequence contained in the macro will be automatically reproduced within the active screen. Keep in mind that macros can only be triggered within the connection they were created.

If you would like to stop the macro reproduction before it finishes, press the 'Stop Macro Playback' button



 Once pressed, this button will be replaced by the 'Run Macro' button.

Otherwise when the macro reproduction finishes, the 'Stop Macro Playback' button will be replaced by the 'Run Macro' button automatically.

Editing a Macro

On most of the cases you can create, manage and use macros directly following the

instructions above. On occasion, however, you might want to edit the code inside the macro. For that matter we have devoted an exclusive z/Scope section. Learn all about the advanced configuration in the following topic:

- [Macros Settings](#)

3.2.9 MS Office Integration

z/Scope incorporates Microsoft® Office integration to allow the user to experience the convenience of data exportation between applications.

z/Scope offers integration with Word and Outlook® applications from the Microsoft® Office suit.

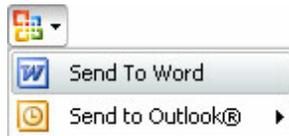
To get more information about how data exportation between z/Scope and these applications works, read the following sections:

- [Exporting to Microsoft Word](#)
- [Exporting to Microsoft Outlook](#)

3.2.9.1 Exporting to Microsoft® Word

Instead of using the basic copy/paste technic, z/Scope enables the user to export the active screen directly as a Microsoft® Word document.

In the main toolbar, click on the arrow next to the 'MS Office' button and select the 'Send to Word' option.



A new instance of Microsoft® Word will be launched containing the terminal emulation screen text.

If you want to export only a portion of the active screen, select the target text using the mouse and then select the 'Send to Word' option as explained before. Otherwise, the whole screen will be exported.

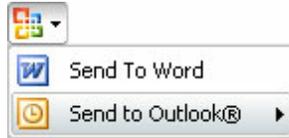
Related Topics

- [Exporting to Microsoft Outlook](#)

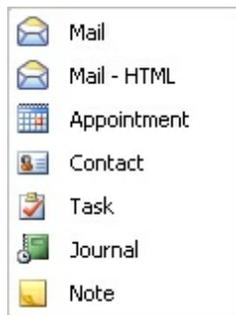
3.2.9.2 Exporting to Microsoft® Outlook

Instead of using the basic copy/paste method, z/Scope enables the user to export the active screen directly as a Microsoft® Outlook element.

In the main toolbar, click on the arrow button next to the 'MS Office' button and select the 'Send to Outlook' option.



A list of different Outlook® documents will be unfolded. Click on the one you want and a new instance of Microsoft® Outlook will be launched, containing the text of the terminal emulation screen placed in the type of document selected.



If you want to export only a portion of the active screen, select the target text using the mouse and then select the 'Send to Outlook' option as explained before. Otherwise, the whole screen will be exported.

Related Topics

- [Exporting to Microsoft Word](#)

3.2.10 Navigators

Navigators are a very valuable tool that allows users to define automated navigation sequences between frequently visited screens. In this way you can, for example, automate in a simple click the welcome and login procedure of a host you access frequently.

Navigators are in many ways similar to Macros; one major difference is that Navigators are specifically tailored for a precise group of screens, thus operating in much more restricted circumstances.

Navigators, like Auto-fills and HotSpots, can only be used within the connection they were defined. Moreover, Navigators only become available for triggering when the active screen displayed is the exact same one that was established as the starting point for the

automated navigation sequence when the Navigator was created.

To learn how to create and run your own Navigators, see the next topics:

- [Creating a Navigator](#)
- [Running a Navigator](#)

Related Topics

[Navigators Settings](#)

3.2.10.1 Creating a Navigator

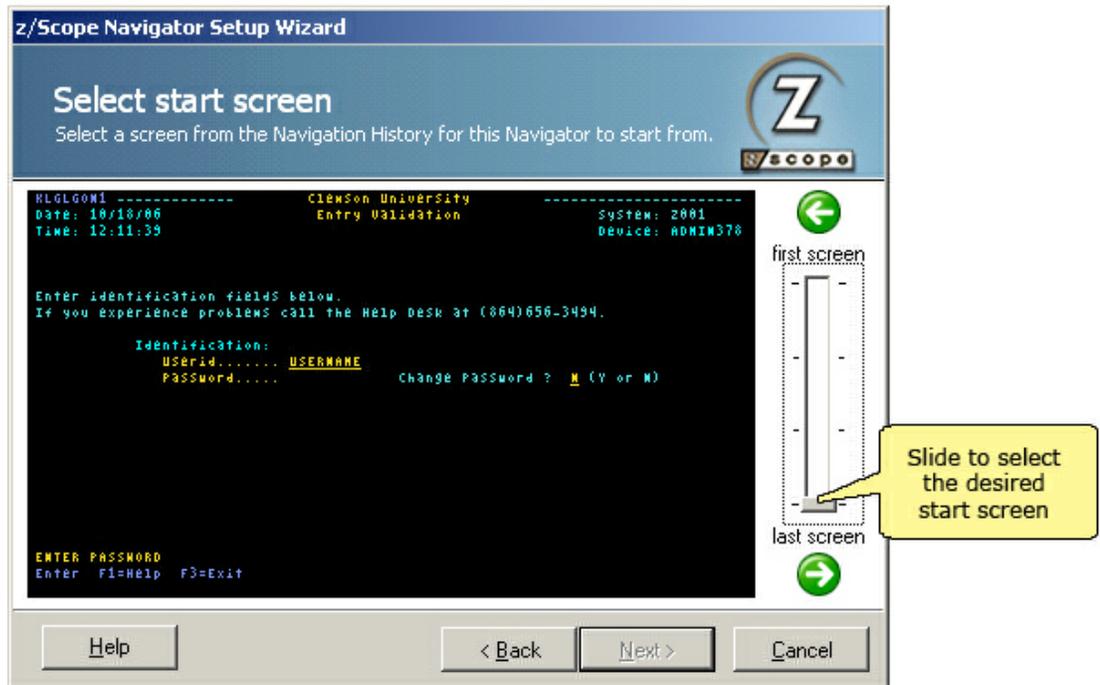
Navigators are easily created through the 'Navigator Wizard' button in the main connection toolbar.



First of all, you will be presented with the 'Navigator Wizard' welcome screen.

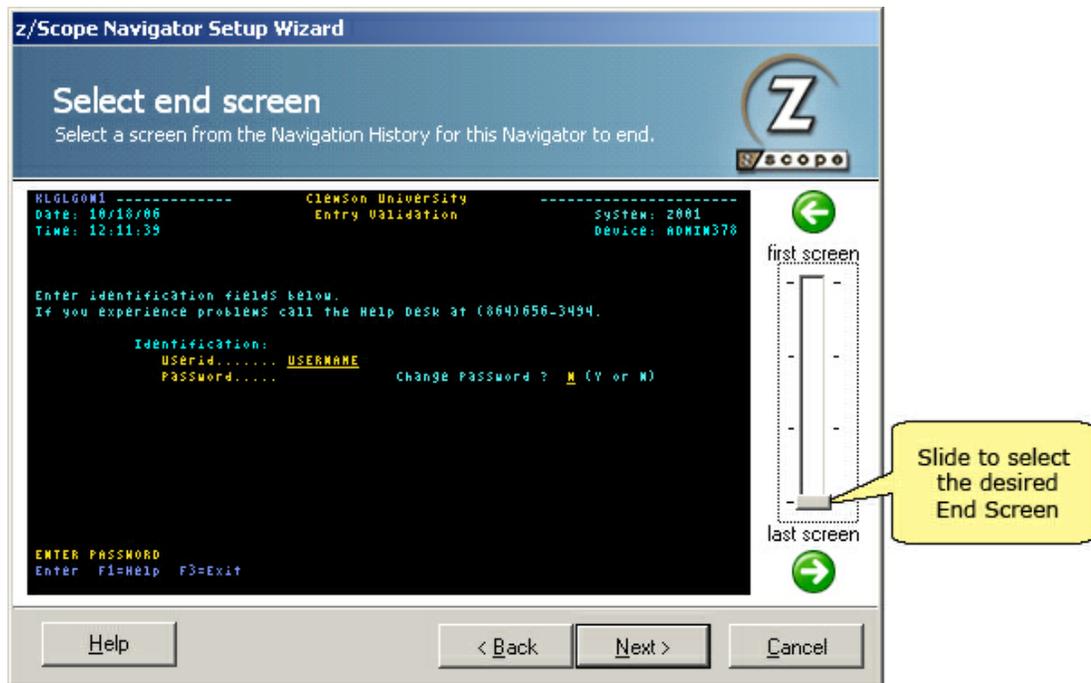


Click on 'Next' to go to the next step, in which you will have to specify the start screen for the automated navigation sequence.



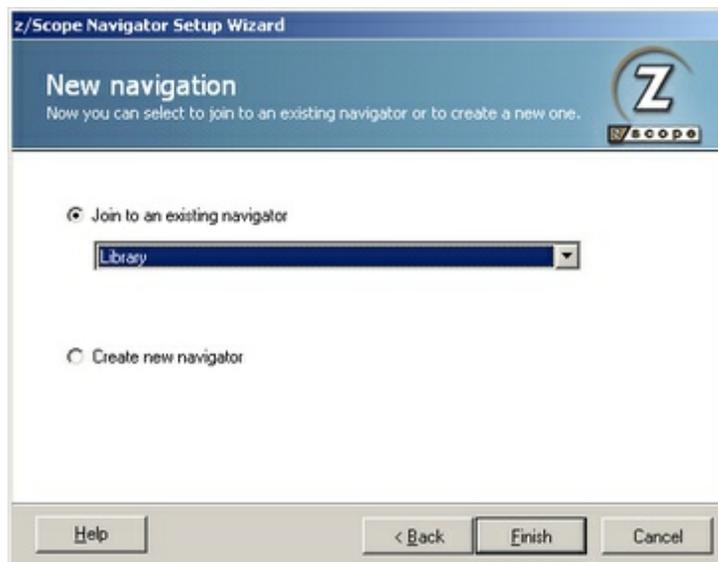
Scroll through the navigation history using the slide bar until you locate the desired start screen. You can also use the arrows to select the desired screen. Click on 'Next' when done.

Now proceed in the same way for the desired End screen.



Once you are done selecting both Start and End screens, click on 'Next' to go to the next step, in which you can choose to create a new Navigator or join it to an existing one in the following dialog:

- ! This screen will not appear if you have no other Navigators configured. You will be presented with the Visualization Settings screen instead.



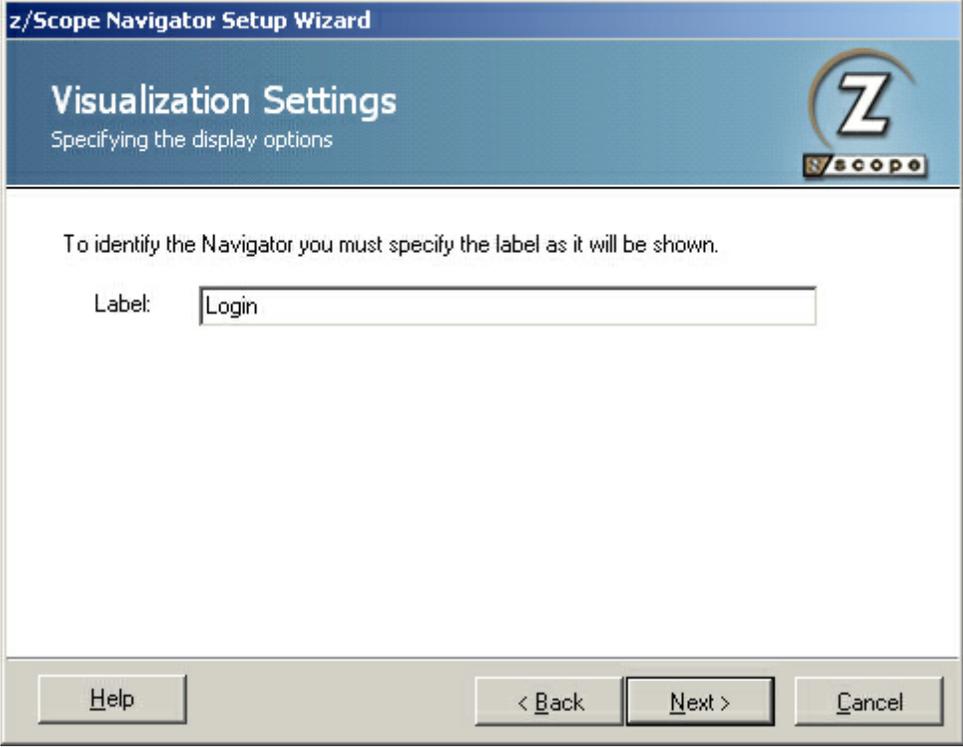
Join To An Existing Navigator

In this list you will find available Navigators that have the same end screen as the one you are creating. Choose the one the current Navigator will be joined with and

then press Finish to exit the Navigator Wizard.

Create New Navigator

By choosing this option you will create a new Navigator with a new name and label. Press Next to continue and enter Visualization Settings.

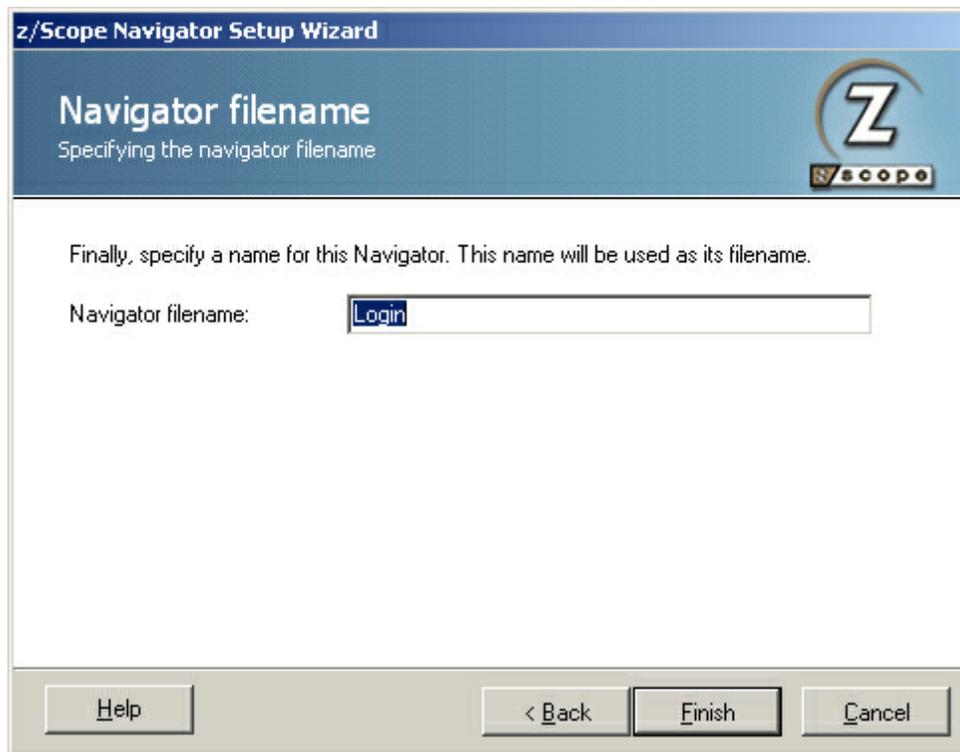


The screenshot shows a dialog box titled "z/Scope Navigator Setup Wizard" with a sub-header "Visualization Settings" and the subtitle "Specifying the display options". The z/SCOPE logo is in the top right corner. The main text reads: "To identify the Navigator you must specify the label as it will be shown." Below this is a "Label:" label followed by a text input field containing the word "Login". At the bottom, there are three buttons: "Help", "< Back", and "Next >", and a "Cancel" button on the far right.

Label

In this field you must enter a caption for the Navigator's button. This label name will be seen in the drop down menu of the Play Navigator button in the main toolbar and in the proper navigation button in the Navigator toolbar.

Finally, click on 'Next' to go to the last step of the Navigator creation process.



Navigator Filename

In this field you must enter a file name for the Navigator you just created.

Click on 'Finish' to confirm the new Navigator creation. Otherwise, click on 'Cancel' to discard the whole procedure.

To learn how to use Navigators once they have been created, see the next topic:

- [Running a Navigator](#)

Related Topics

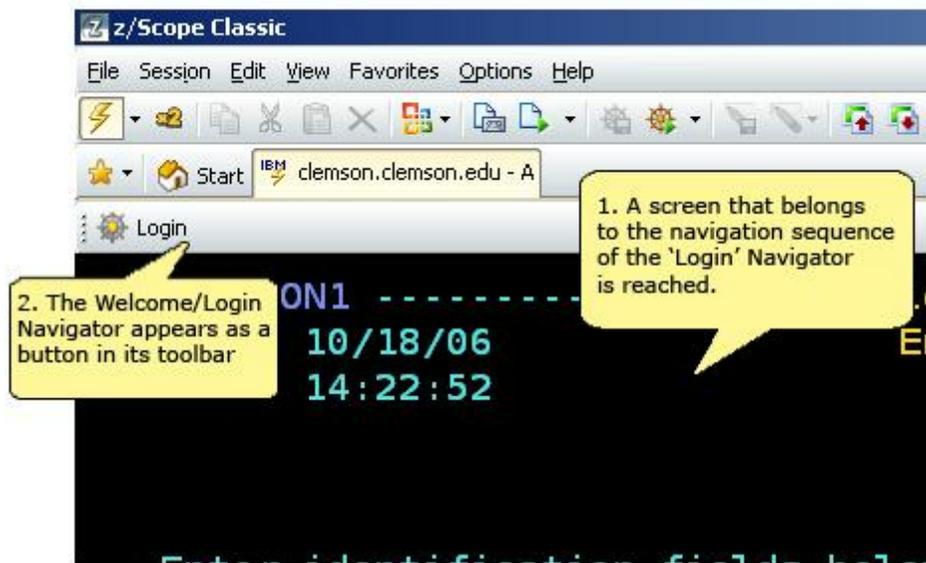
[Navigators Settings](#)

3.2.10.2 Running a Navigator

There are three ways in which a Navigator can be run:

1. From a button in a user-defined toolbar

If the Navigator is configured to work as a button in a toolbar, it will automatically appear in its corresponding toolbar at the top of the emulation display (as shown in the screenshot below) when reaching any given screen that belongs to the navigation sequence.



In order for Navigators to behave in this way, the 'Context Selection' option in the ['General' tab of the Navigation Settings](#) must be unchecked. This behaviour is also set by default when creating a Navigator using the 'Navigator Wizard'. See [Creating a Navigator](#).

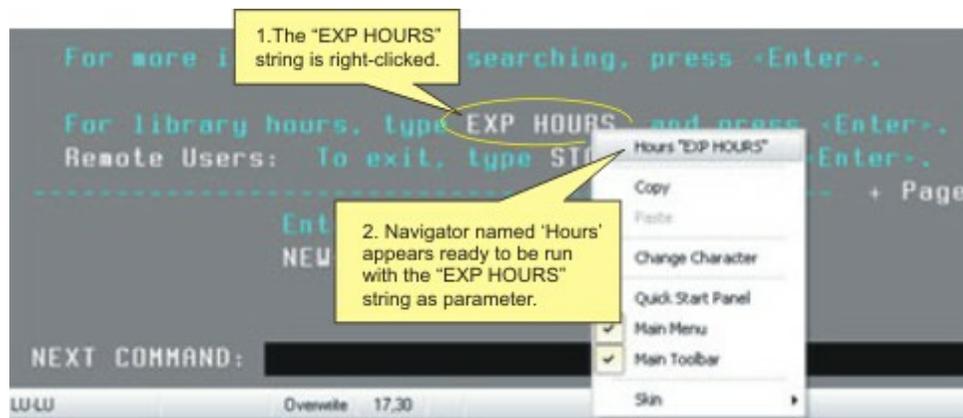
2. From the main toolbar



Navigators can alternatively be run from the 'Run Navigator' button in the main toolbar. This method is available under the same circumstances that were described in method number one. If you click on the arrow next to this button, a list of all available Navigators for the current screen will be unfolded.

3. From a context pop-up menu

If the Navigator is configured to work as a context pop-up menu, it will be available as a menu item (as shown in the screenshot below) when you right-click on any text contained in a screen that is part of the navigation sequence.



In the screenshot above, you can see that when the text string "EXP HOURS" is right-clicked, the Navigator named 'Hours' becomes available, together with the selected string ready to be passed as a parameter. When the Navigator is then executed, every text field that has been set using the 'Switch to Parameter' button in the 'Screen Fields' tab of the [Navigators Settings](#) will be automatically filled using the selected string.

In order for Navigators to behave in this way, the 'Context Selection' option in the 'General' tab of the [Navigator Settings](#) must be checked.

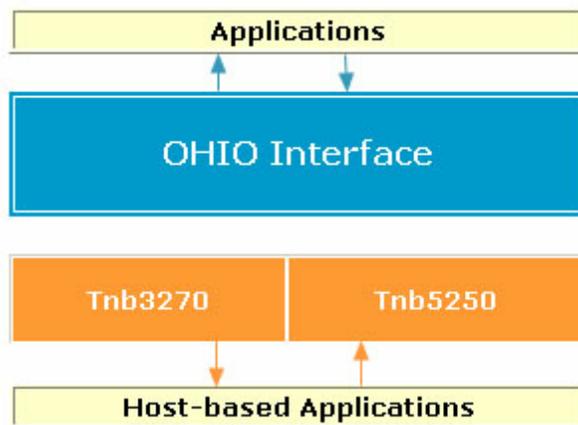
Related Topics

[Creating a Navigator](#)
[Navigators Settings](#)

3.2.11 OHIO

Open Host Interface Objects (OHIO) is an IBM and Attachmate inspired object-oriented host access API for software implementing TN3270 and TN5250 protocols. Vendor neutral and submitted to the IETF in 1998 as an Internet standard.

OHIO addresses the need for a standardized advanced programming interface that can access the host data. OHIO does not modify the TN3270/TN5250 protocol or datastream but instead provides a common access method to that data once it arrives at the client. It uses an Object Oriented approach to divide the data into logical objects, and provides methods on those objects to allow standard access to the data.



Some of the OHIO's advantages are:

- You can run an OHIO application without having an emulation session running.
- OHIO has all the benefits of the object oriented programming paradigm because it was designed as an object oriented API.
- You can concentrate on the application functions, without worrying about structure packing details or parameter command codes.

To learn more about OHIO, take a look at the [OHIO Reference](#).

3.2.12 HLLAPI

HLLAPI is a standard programming interface that allows programmatic access to a TN3270 and TN5250 host emulator sessions. Functions are provided for reading host screen data (such as the characters and attributes), for sending keystrokes, and for performing other emulator-related functions.

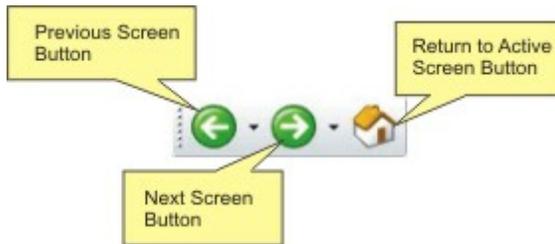
The EHLLAPI interface is a single call-point interface. There is a single callable API through which all EHLLAPI functions are requested. On each call to the interface the application provides a function number which identifies the function requested, a pointer to a data buffer, a pointer to the length of the data buffer, and a pointer to a return code.

WinHLLAPI is based on the familiar EHLLAPI.API. It encompasses all of its existing functionality and adds extensions that take advantage of the Windows message-driven environment.

To learn more about HLLAPI, take a look at the [HLLAPI Reference](#).

3.2.13 Screen History

The Screen History feature keeps track of all visited screens, allowing you to go back and forth at any time in the same way as most popular Internet browsers do. The Screen History is accessible from the main toolbar.

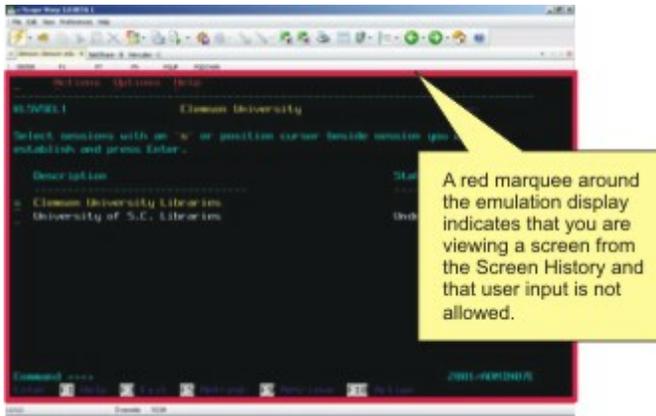


When no screens have been visited you will note that both the 'Previous' and 'Next' buttons are disabled. Once you begin navigating through different screens, these buttons will become enabled, allowing you to consult them.

An important aspect of this feature is the possibility to jump directly to a particular screen using a thumbnail system located in drop-down list that unfolds when you click on the arrows that appear to the right of both navigation buttons.



To have access to a screen in particular, just click on its corresponding thumbnail. The screen will then be loaded, restricted by a red marquee indicating that it is not the active screen (it belongs to the Screen History) and that no input is allowed.



To go back to the active screen, all you have to do is click on the 'Return to Active Screen' button.

Related Topics:

- [Configure how many screens you want to keep in the Navigation History in the 'Misc' tab in Environment Settings](#)

3.2.14 Screen Styles

Screen Styles give the user the possibility to freely modify the aspect of the emulation display. You can customize a wide variety of characteristics such as font format and size, cursor appearance and behaviour, color schemes, etc.

To learn more about Screen Styles, go to the next topic:

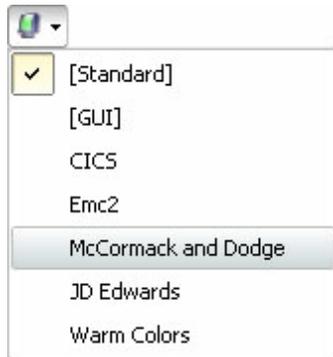
- [Using Screen Styles](#)

Related Topics

- [Screen Styles Settings](#)

3.2.14.1 Using Screen Styles

To change the Screen Style for the current emulation, click on the 'Screen Styles' button in the main toolbar. A list containing all available Screen Styles will be unfolded.



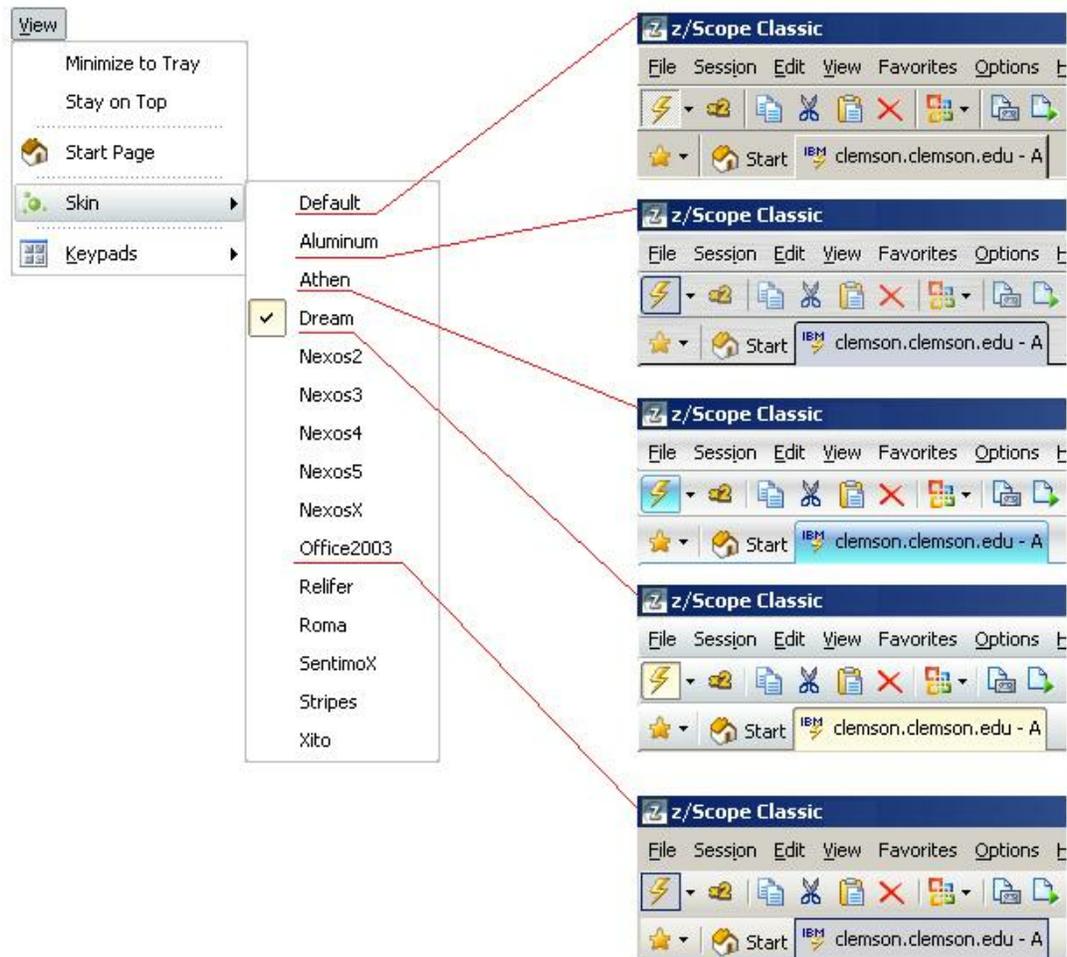
Just click on the one you want and it will be instantly loaded.

Related Topics

- [Screen Styles Settings](#)

3.2.15 Using z/Scope Skins

Z/Scope is shipped with many slick skins to change the appearance of the interface elements, such as menus, toolbars and tabs. To change the active skin, select a different one from the 'Skins' list in the 'View' menu.



3.2.16 Security and Encryption

With z/Scope you can connect securely via the SSL protocol. SSL stands for Secure Sockets Layer and was originally developed by Netscape for transmitting private information and documents over the Internet.

SSL is based on a private key encryption system. Many web sites use this protocol to transmit confidential user information such as credit card numbers.

An SSL digital certificate is an electronic file that uniquely identifies individuals and servers. Digital certificates allow the client to authenticate the server prior to establishing an SSL session.

Typically, digital certificates are signed by an independent and trusted third party to ensure their validity. The "signer" of a digital certificate is known as a Certification Authority (CA), such as VeriSign®.

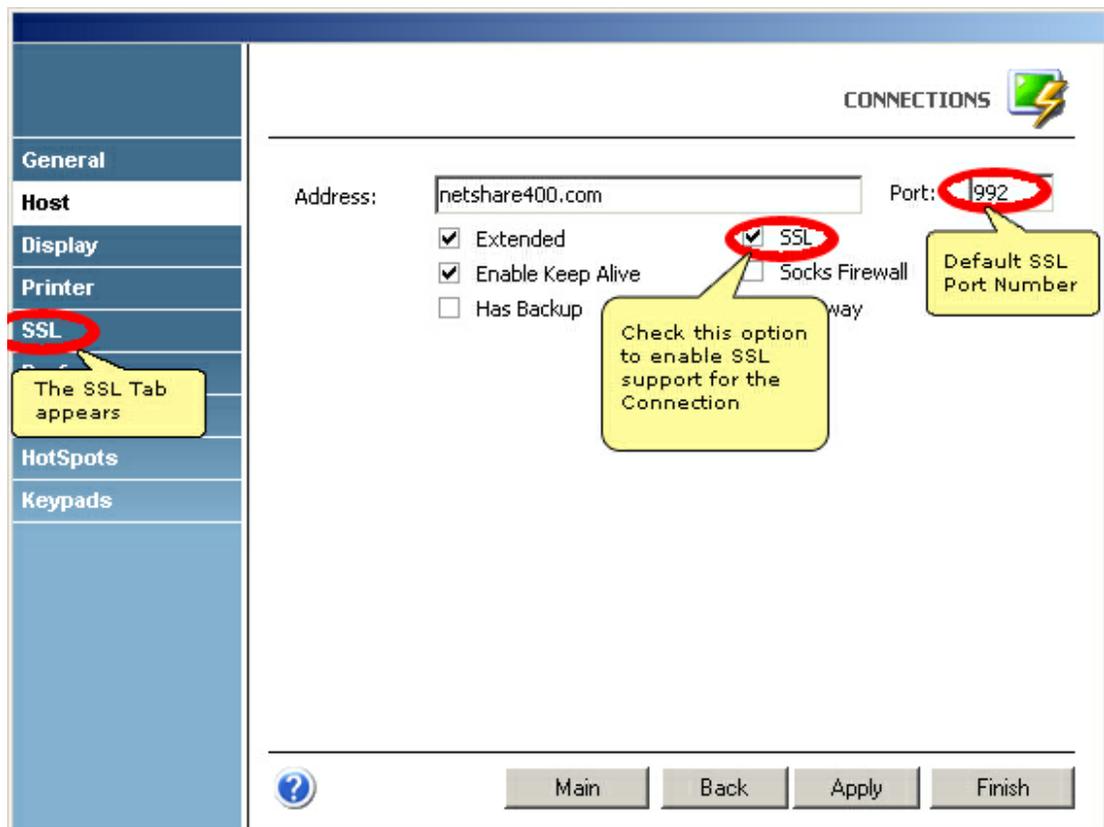
For more information about how to work with SSL in z/Scope, read the following topics:

- [Using SSL](#)

3.2.16.1 Using SSL

When connecting to a host that supports SSL, in order to take advantage of this technology you must activate the SSL option in the [Connection Settings](#). To achieve this, follow these steps:

1. Go to the 'Settings' dialog.
2. Click on the 'Connections' icon.
3. Select the connection that you want to activate SSL encryption for.
4. Click on the 'Host' tab.
5. Check the 'SSL' option. Note that when you do this two things will happen: First, the port number will be automatically set to 992, the default port number for the SSL protocol. Secondly, the 'SSL' tab will appear in the connection settings dialog.

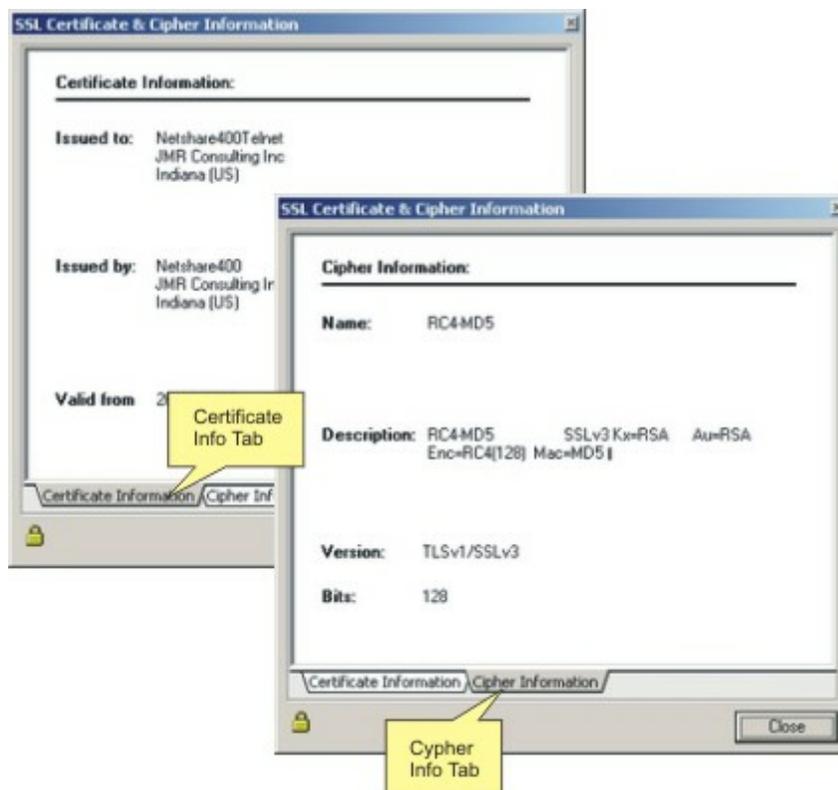


For a detailed description of the options available in the 'SSL' tab, see [SSL tab in the Connection Settings for 3270 and 5250](#) and [SSL tab in the Connection Settings for vt](#).

Now that SSL is enabled, this will be informed in the 'Session Status' panel of the status bar.



If you double-click on that area, the 'SSL Certificate and Cipher' information dialog will be displayed. Use the tabs to switch between the Certificate and Cipher information page.



3.3 Setting up a Connection

The Connection Wizard helps you set up a new connection by following a few simple steps. You can launch the 'Connection Wizard' from the 'File' menu.



Also you can access the Connection Setup Wizard through the 'Settings' dialog. You will first see the 'Welcome' dialog.



Click on 'Next' to start the connection creation procedure. You can go back to the previous step at any time by clicking 'Back'. Also, you can click 'Cancel' at any time during the procedure to exit the wizard.

To learn about the different type of hosts you can connect to using z/Scope, refer to the following sections.

- [Setting up a UNIX/VT Connection](#)
- [Setting up an IBM Mainframe or AS/400 connection](#)
- [Setting up an FTP connection](#)

For a detailed description on how to further customize the connections you have

created, see the following topics:

- [Editing a Unix/VT connection](#)
- [Editing an IBM Mainframe or AS/400 Connection](#)
- [Editing an FTP connection](#)

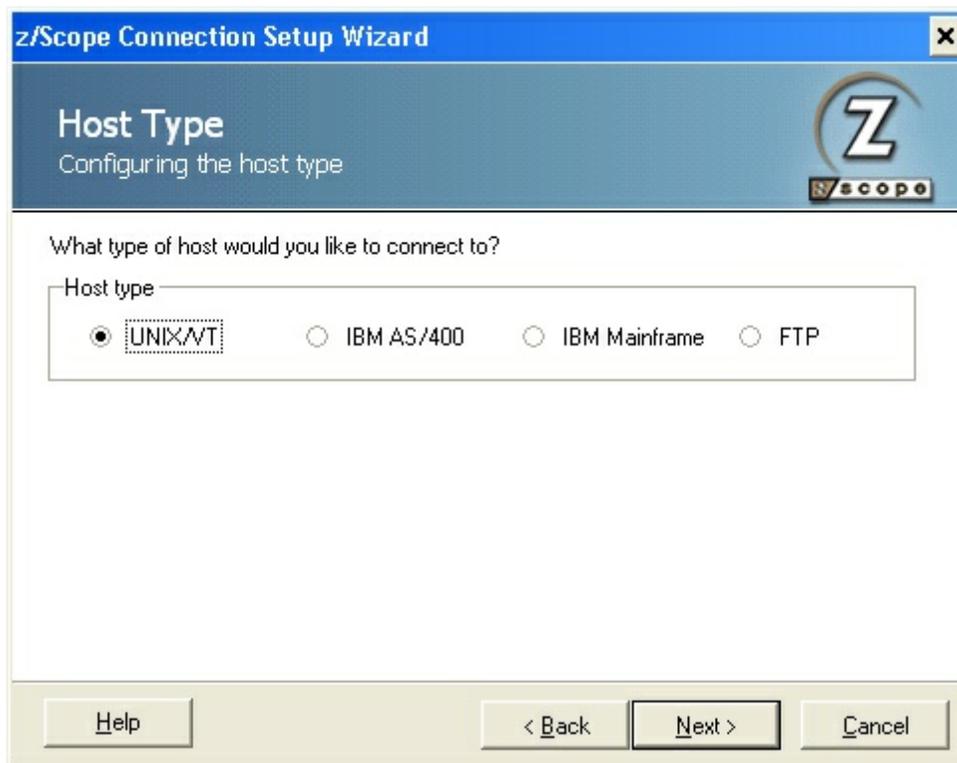
Copyright © 2002-2012, Cybele Software Inc. All rights reserved.

3.3.1 Setting up a UNIX/VT/SSH Connection

This page will guide you to create a UNIX/VT/SSH connection.

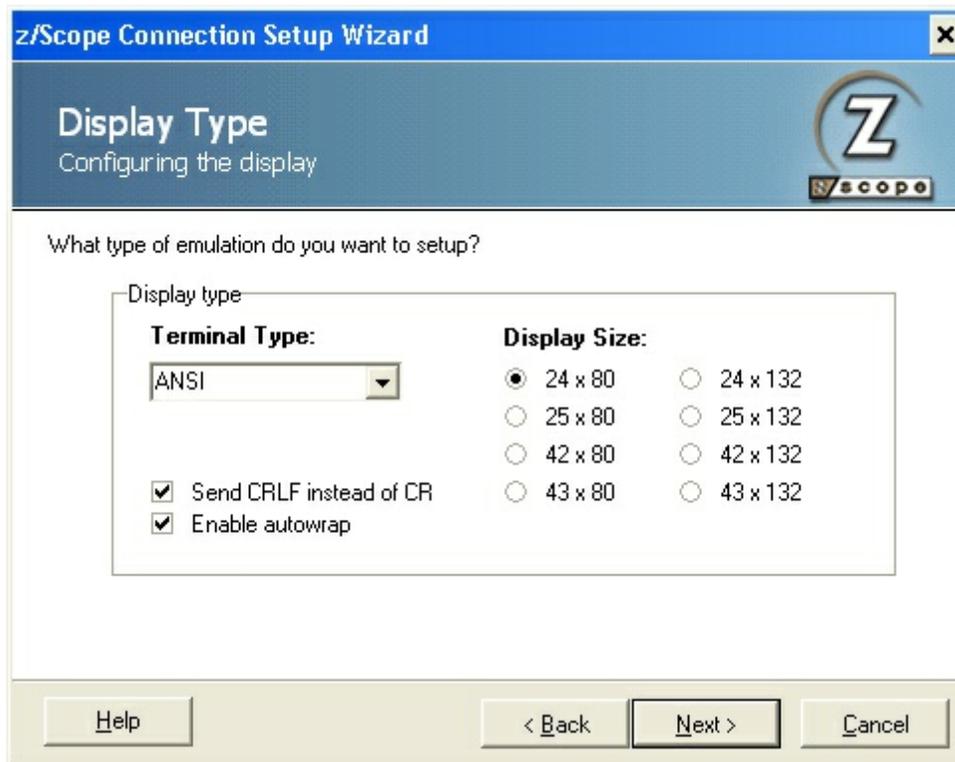
Host Type

Once in the second screen of the Connection Setup Dialog, select 'UNIX/VT' as the host type and click 'Next'.



Display Type

In the 'Display Type' screen, specify the display type and size. Click 'Next' to continue.



Host Address

In this step you must enter the URL or IP address of the host you want to connect to.

If you would like to specify a port different than the default Telnet port (23), uncheck the 'Default Port' option and then enter a port number in the 'Telnet' field.



The screenshot shows a window titled "z/Scope Connection Setup Wizard" with a sub-header "Host Address" and the text "Providing the host address". The z/Scope logo is in the top right. The main area contains the instruction: "Fill in the Host Address field. This address corresponds to the DNS name or IP address assigned to the host." Below this is a text input field labeled "Host Address:". To the right of the input field is a "Telnet:" label and a small input field containing the number "23". Below the input field are three checkboxes: "Default port", "HTTP Gateway", and "SSH". At the bottom of the window are four buttons: "Help", "< Back", "Next >", and "Cancel".

There are other options you can check, and each one has different settings: [SSH](#) and [HTTP Gateway](#).

3.3.1.1 Connection with SSH

This page will guide you to create a UNIX/VT/SSH connection with SSH.

Once in the Host Address screen of the Connection Setup Dialog, select enter the host, the port, check SSH and click 'Next'.



The image shows a screenshot of the 'z/Scope Connection Setup Wizard' window. The title bar reads 'z/Scope Connection Setup Wizard'. The main heading is 'Host Address' with the subtitle 'Providing the host address'. The z/Scope logo is in the top right corner. Below the heading, there is a text instruction: 'Fill in the Host Address field. This address corresponds to the DNS name or IP address assigned to the host.' The 'Host Address' field contains the text 'MyHostAddress.com'. To the right of this field is a 'Telnet:' field containing the number '22'. Below these fields are three checkboxes: 'Default port' (unchecked), 'HTTP Gateway' (unchecked), and 'SSH' (checked). At the bottom of the window are four buttons: 'Help', '< Back', 'Next >', and 'Cancel'.

Only if you have marked the SSH checkbox, following the 'Host Address' screen you will be presented with the following screen. Enter the username and password to identify you in the connection.

- ! When selecting to configure an SSH connection, you will not be able to use a Gateway for this.



Additionally you can specify a private key file by marking the 'Private Key Field' checkbox and selecting the file with the '.' button. Then press 'Next'.

Finally, enter a name for the new connection.



If you have accessed the Connection Setup Wizard through the main toolbar or main menu, you can leave the 'Connect to Host' option checked to connect right after the connection is created.

To create the connection click 'Finish', or press 'Cancel' to discard the whole procedure.

Related items:

- [Editing a UNIX/VT Connection](#)
- [Creating a Connection](#)

3.3.1.2 Connection with HTTP Gateway

This page will guide you to create a UNIX/VT/SSH connection with a HTTP Gateway.

Once in the Host Address screen of the Connection Setup Dialog, select enter the host, the port and check the box to enable z/Scope Anywhere.



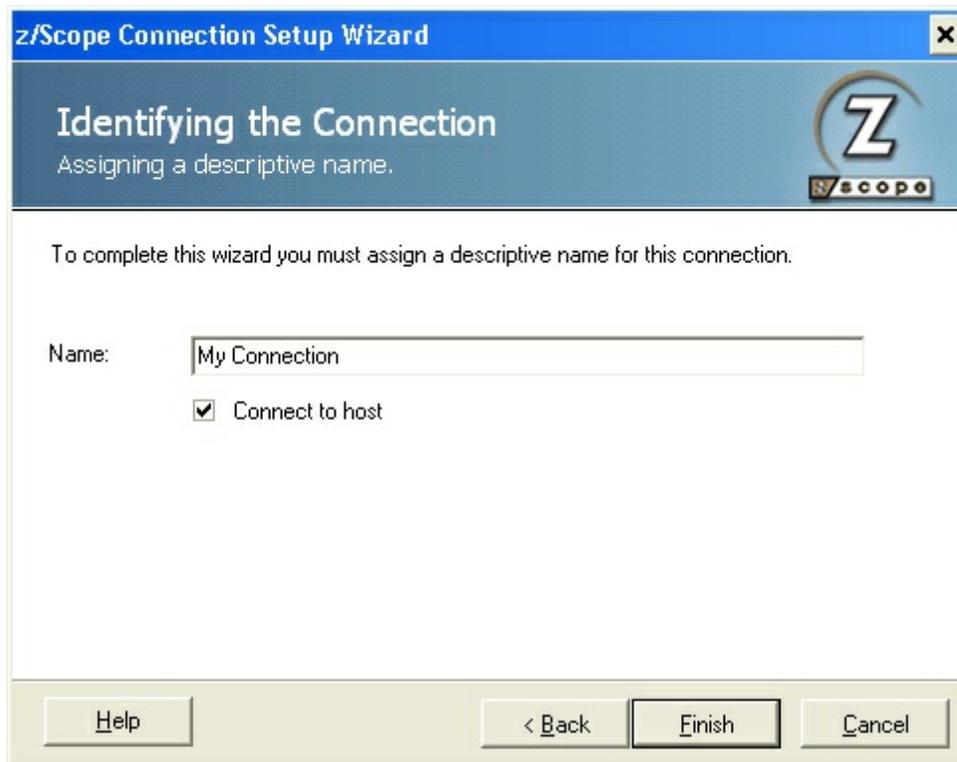
The image shows a screenshot of the 'z/Scope Connection Setup Wizard' window, specifically the 'Host Address' step. The window title is 'z/Scope Connection Setup Wizard' and it features the z/Scope logo. The main heading is 'Host Address' with the subtitle 'Providing the host address'. Below this, there is a text instruction: 'Fill in the Host Address field. This address corresponds to the DNS name or IP address assigned to the host.' The form contains several fields and options: a 'Host Address' text box with 'MyHostAddress.com' entered; a 'Default port' checkbox which is unchecked; a 'Telnet' text box with '23' entered; an 'HTTP Gateway' checkbox which is checked, with a corresponding text box containing 'http://MyGateway.com'; and an 'SSH' checkbox which is unchecked. At the bottom of the window, there are four buttons: 'Help', '< Back', 'Next >', and 'Cancel'.

A new box will be enabled, so you can enter the HTTP Gateway you wish to use.

 You must enter "http://", otherwise you will not be able to continue.

Then press 'Next'.

Finally, enter a name for the new connection.



If you have accessed the Connection Setup Wizard through the main toolbar or main menu, you can leave the 'Connect to Host' option checked to connect right after the connection is created.

To create the connection click 'Finish', or press 'Cancel' to discard the whole procedure.

Related items:

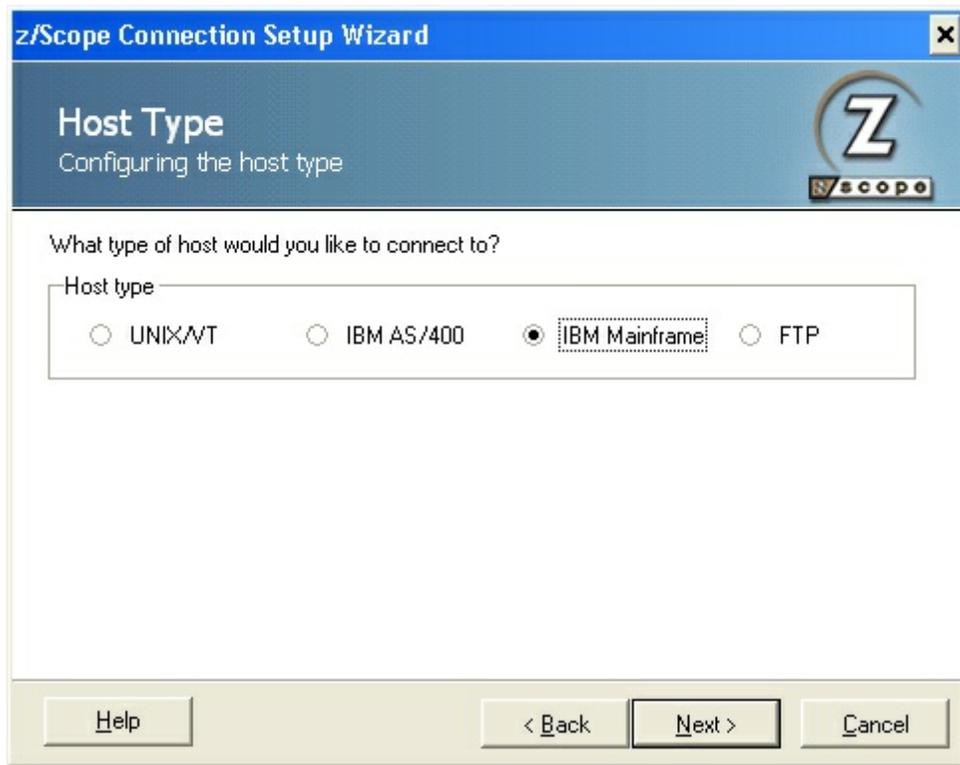
- [Editing a UNIX/VT Connection](#)
- [Creating a Connection](#)

3.3.2 Setting up an IBM Mainframe or AS/400 Connection

This page will guide you to create an IBM Mainframe or AS/400 Connection.

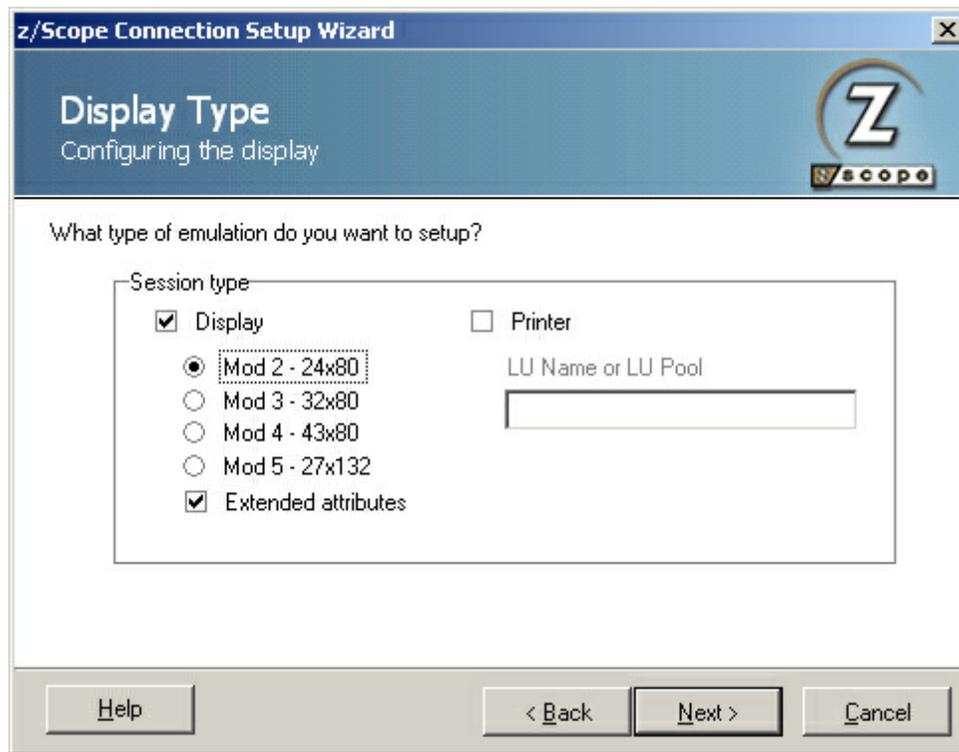
Host Type

Once in the second screen of the Connection Setup Dialog, select IBM Mainframe or IBM AS/400 as the Host Type and click 'Next'.



Display Type

Specify the display type. The list of available modes will vary according to the host type you selected in the previous step.



Mark both the 'display' and 'printer' checkbox if you would like to configure printer and display under the same connection name (this will allow you to configure for example simultaneous autoconnection).

Mark only the 'Printer' checkbox if you need to configure a printer-only connection to the host. If you mark the 'Printer' checkbox, specify the device name or LU Pool.



Only z/Scope Classic, Web-to-Host and Workbench Editions have Printer Emulation capabilities.

Click 'Next' to continue.

Host Address

In this step you must enter the URL or IP address of the host you want to connect to.

If you would like to specify a port different than the default Telnet port (23), uncheck the 'Default Port' option and then enter a port number in the 'Telnet' field.



The screenshot shows a window titled "z/Scope Connection Setup Wizard" with a sub-header "Host Address" and the text "Providing the host address". The z/Scope logo is in the top right. The main area contains the instruction: "Fill in the Host Address field. This address corresponds to the DNS name or IP address assigned to the host." Below this is a "Host Address:" label followed by a text input field. To the right of the input field is a "Telnet:" label followed by a small input field containing the number "23". Below the input fields are three checkboxes: "Default port", "HTTP Gateway", and "SSH". At the bottom of the window are four buttons: "Help", "< Back", "Next >", and "Cancel".

There are other options you can check, and each one has different settings: [SSH](#) and [HTTP Gateway](#).

3.3.2.1 Connection with SSH

This page will guide you to create a UNIX/VT/SSH connection with SSH.

Once in the Host Address screen of the Connection Setup Dialog, select enter the host, the port, check SSH and click 'Next'.



The image shows a screenshot of the 'z/Scope Connection Setup Wizard' window. The title bar reads 'z/Scope Connection Setup Wizard'. The main heading is 'Host Address' with the subtitle 'Providing the host address'. The z/Scope logo is in the top right corner. Below the heading, there is a text instruction: 'Fill in the Host Address field. This address corresponds to the DNS name or IP address assigned to the host.' The 'Host Address' field contains the text 'MyHostAddress.com'. To the right of this field is a 'Telnet:' field containing the number '22'. Below these fields are three checkboxes: 'Default port' (unchecked), 'HTTP Gateway' (unchecked), and 'SSH' (checked). At the bottom of the window are four buttons: 'Help', '< Back', 'Next >', and 'Cancel'.

Only if you have marked the SSH checkbox, following the 'Host Address' screen you will be presented with the following screen. Enter the username and password to identify you in the connection.

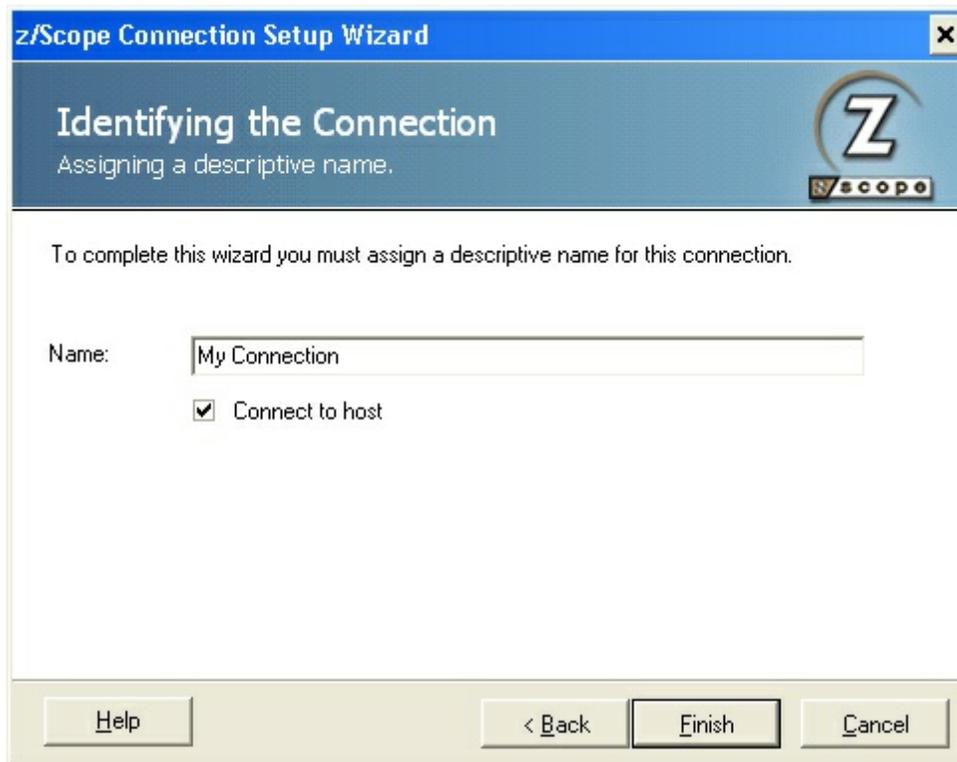
 When selecting to configure an SSH connection, you will not be able to use a Gateway for this.



The image shows a Windows-style dialog box titled "z/Scope Connection Setup Wizard" with a close button (X) in the top right corner. The main heading is "SSH Authentication" with the subtitle "Authenticating the client on your server." and the z/Scope logo. The text inside reads: "You can define an user name and password as well as a private key to be used for authenticating the client." There are two options: "Password Authentication" which is checked, and "Private Key File" which is unchecked. Under "Password Authentication", there are two text input fields labeled "Username:" and "Password:". Under "Private Key File", there is a text input field labeled "Key File:" followed by a browse button (three dots). At the bottom, there are four buttons: "Help", "< Back", "Next >", and "Cancel".

Additionally you can specify a private key file by marking the 'Private Key Field' checkbox and selecting the file with the '.' button. Then press 'Next'.

Finally, enter a name for the new connection.



If you have accessed the Connection Setup Wizard through the main toolbar or main menu, you can leave the 'Connect to Host' option checked to connect right after the connection is created.

To create the connection click 'Finish', or press 'Cancel' to discard the whole procedure.

Related items:

- [Editing a UNIX/VT Connection](#)
- [Creating a Connection](#)

3.3.2.2 Connection with HTTP Gateway

This page will guide you to create an IBM Mainframe or AS/400 Connection with a HTTP Gateway.

Once in the Host Address screen of the Connection Setup Dialog, select enter the host, the port and check the box to enable the HTTP Gateway.



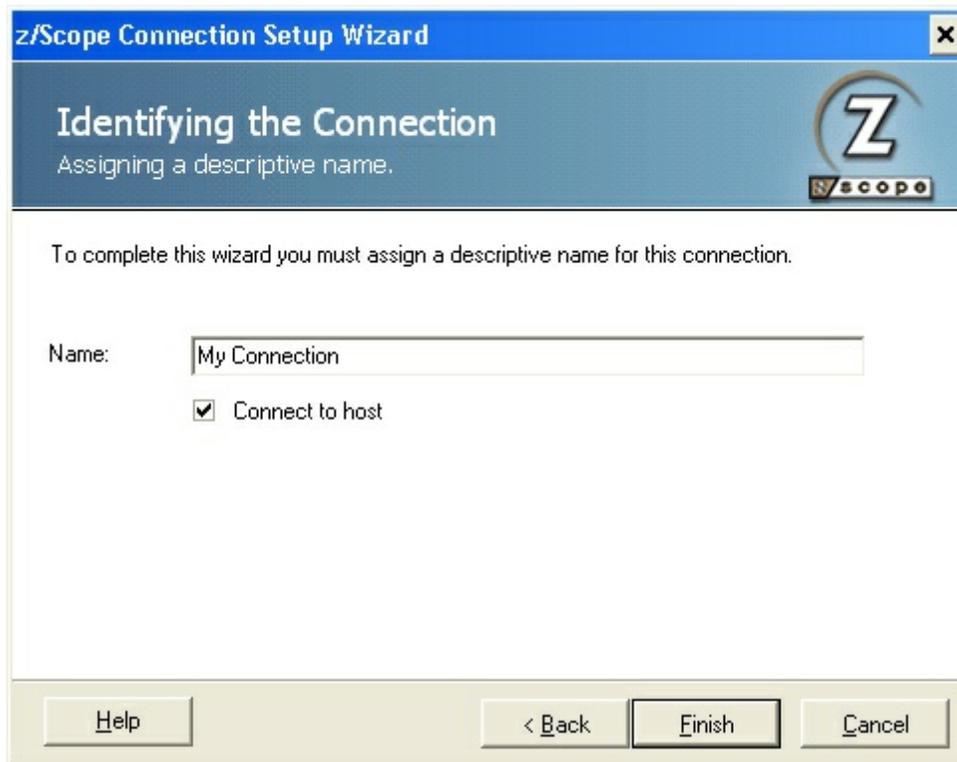
The image shows a screenshot of the 'z/Scope Connection Setup Wizard' window, specifically the 'Host Address' step. The window title is 'z/Scope Connection Setup Wizard' and it features the z/Scope logo. The main heading is 'Host Address' with the subtitle 'Providing the host address'. Below this, there is a text instruction: 'Fill in the Host Address field. This address corresponds to the DNS name or IP address assigned to the host.' The form contains several fields: a 'Host Address' field with the value 'MyHostAddress.com', a 'Default port' checkbox which is unchecked, a 'Telnet' field with the value '23', and an 'HTTP Gateway' checkbox which is checked. The 'HTTP Gateway' field contains the value 'http://MyGateway.com'. At the bottom of the window, there are four buttons: 'Help', '< Back', 'Next >', and 'Cancel'.

A new box will be enabled, so you can enter the HTTP Gateway you wish to use.

 You must enter "http://", otherwise you will not be able to continue.

Then press 'Next'.

Finally, enter a name for the new connection.



If you have accessed the Connection Setup Wizard through the main toolbar or main menu, you can leave the 'Connect to Host' option checked to connect right after the connection is created.

To create the connection click 'Finish', or press 'Cancel' to discard the whole procedure.

Related items:

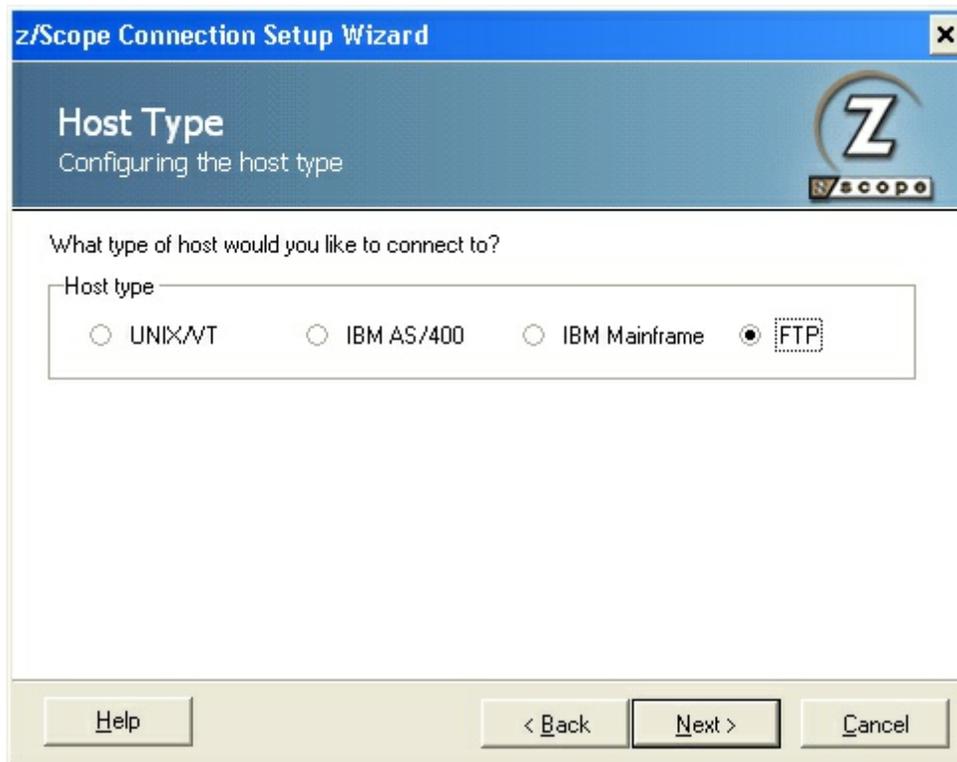
- [Editing an IBM Mainframe or AS/400 Connection](#)
- [Creating a Connection](#)

3.3.3 Setting up an FTP Connection

This page will guide you to create an FTP connection.

Host Type

Once in the second screen of the Connection Setup Dialog, select 'FTP' as the host type and click 'Next'.



Host Adress

In this step you must enter the IP address of the FTP server you want to connect to. If you would like to specify a port different than the default FTP port (21), uncheck the 'Default Port' option and enter a port number in the 'FTP' field.



The image shows a screenshot of the 'z/Scope Connection Setup Wizard' window, specifically the 'Host Address' step. The window title is 'z/Scope Connection Setup Wizard' and it features the z/Scope logo in the top right corner. The main heading is 'Host Address' with the subtitle 'Providing the host address'. Below this, there is a text instruction: 'Fill in the Host Address field. This address corresponds to the DNS name or IP address assigned to the host.' The form contains several fields: 'Host Address' with the value 'MyHostAddress.com', a checkbox for 'Default port' which is unchecked, an 'FTP:' field with the value '21', a checked checkbox for 'z/Scope Anywhere', a 'Friendly Name:' field with the value 'PC Name', and an 'Agent ID:' field with the value '1234-1234-1234-ZA'. At the bottom of the window, there are four buttons: 'Help', '< Back', 'Next >', and 'Cancel'.

If you enable the z/Scope Anywhere function for this connection, two new fields will be enabled, so you can enter the "Friendly name" and the "Agent ID" for the computer that provides you with the remote access to your host. Then press 'Next'.

FTP User Identification

If the host you are connecting to requires user validation, enter a User ID and a Password. If you prefer to log in anonymously, check the 'Anonymous' option. You can also leave all options blank and the system will prompt you for this information when you connect to the host.

z/Scope Connection Setup Wizard

FTP User Identification

Who will be logging in?

You can define a default user identification and password for the ftp connection. These parameters are optional and can be completed while you connect to the host.

User Id:

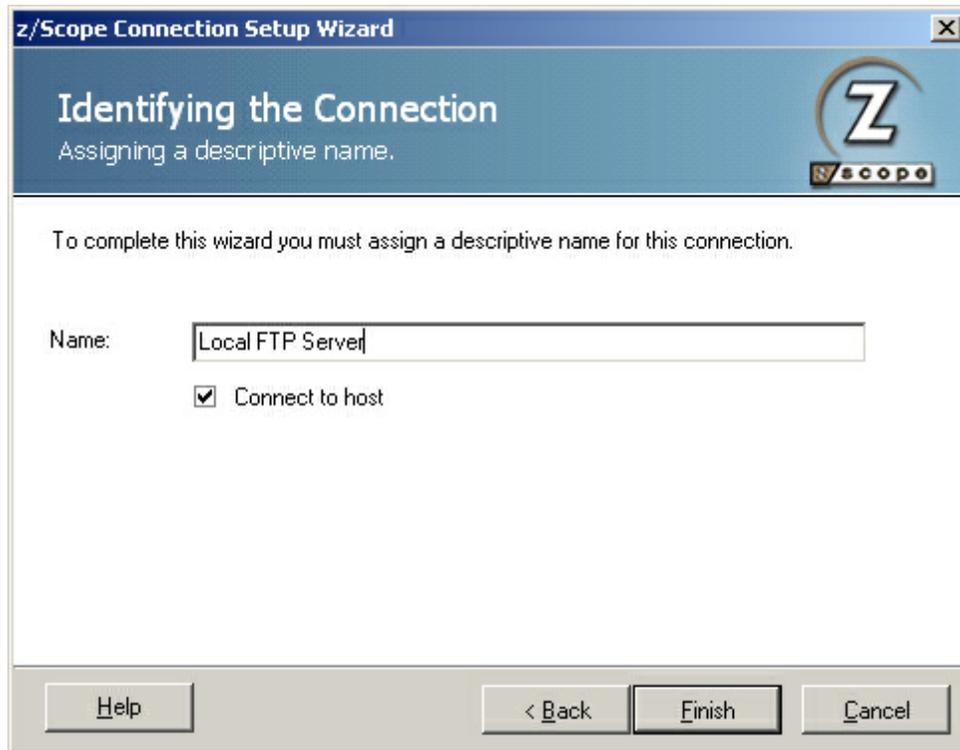
Password:

Anonymous

[Help](#) < [Back](#) [Next](#) > [Cancel](#)

Identifying the Connection

Finally, enter a name for the new connection. If you have accessed the Connection Setup Wizard through the main toolbar or main menu, you can leave the 'Connect to Host' option checked to connect right after the connection is created. To create the new connection, click 'Finish', or click 'Cancel' to discard the whole procedure.



Related items:

- [Editing an FTP Connection](#)
- Creating a Connection

4 Settings

Z/Scope settings and preferences are configured through the 'Settings' dialog. To open this dialog, just click on the 'Settings' button in the main toolbar.



'Settings' button.

Alternatively, you can access from the main menu under 'Options' - 'Configuration Settings':



Click on any of these categories to open the corresponding dialog. These dialogs will be explained in detail throughout the rest of this chapter:

- [Connections](#)
- [Screen Styles](#)
- [HotSpots](#)
- [Keyboard](#)
- [Macros](#)
- [Keypads](#)
- [Navigators](#)
- [Auto-Fills](#)
- [Environment](#)

If you want to modify the settings for the active connection, you can open the corresponding dialog by selecting the category from the list that unfolds when clicking on the arrow next to the 'Settings' button.



This will open the selected settings dialog corresponding to the connection you are currently working with.

Some connection types do not enable Macros or other major functions, so you might see different settings depending on the current connection.

Copyright © 2002-2012, Cybele Software Inc. All rights reserved.

4.1 Connections

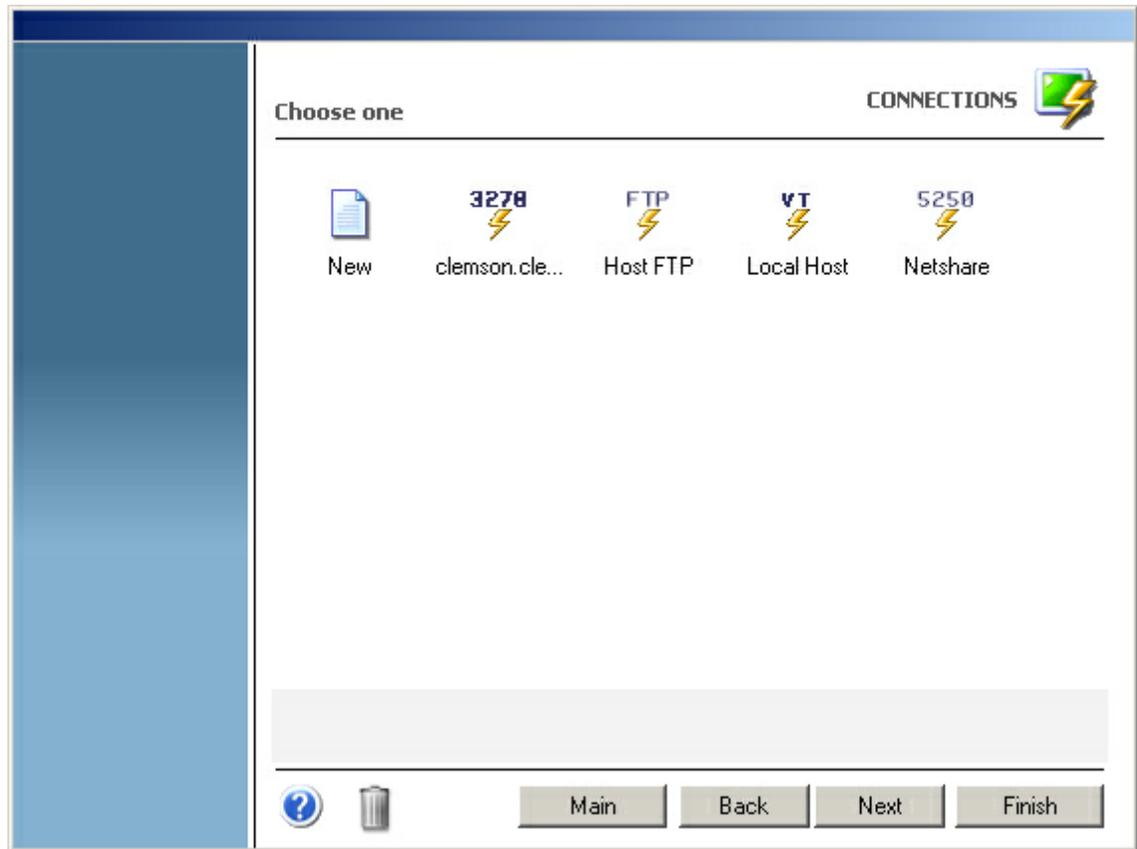
Connections are managed from the 'Connections' icon in the 'Settings' dialog.



The Connections management dialog shows a list of the existing Connections. You will also find the 'New' icon which you can use to create a new Connection from scratch.

To modify the attributes of a previously created Connection, select it using the mouse and then click on 'Next', or just simply double-click on it.

To create a new Connection, double-click on the 'New' icon to launch the Connection Wizard. See



After you select a connection and click on 'Next', you will be presented with a dialog in which you will have access to all the attributes of the Connection you are editing. For a detailed description of these parameters, consult the following sections:

- UNIX/VT Connections
- [Mainframe/AS400 Connections](#)
- [FTP Connections](#)
- [RD Connections](#)
- [VNC connections](#)

4.1.1 Unix/VT Connections

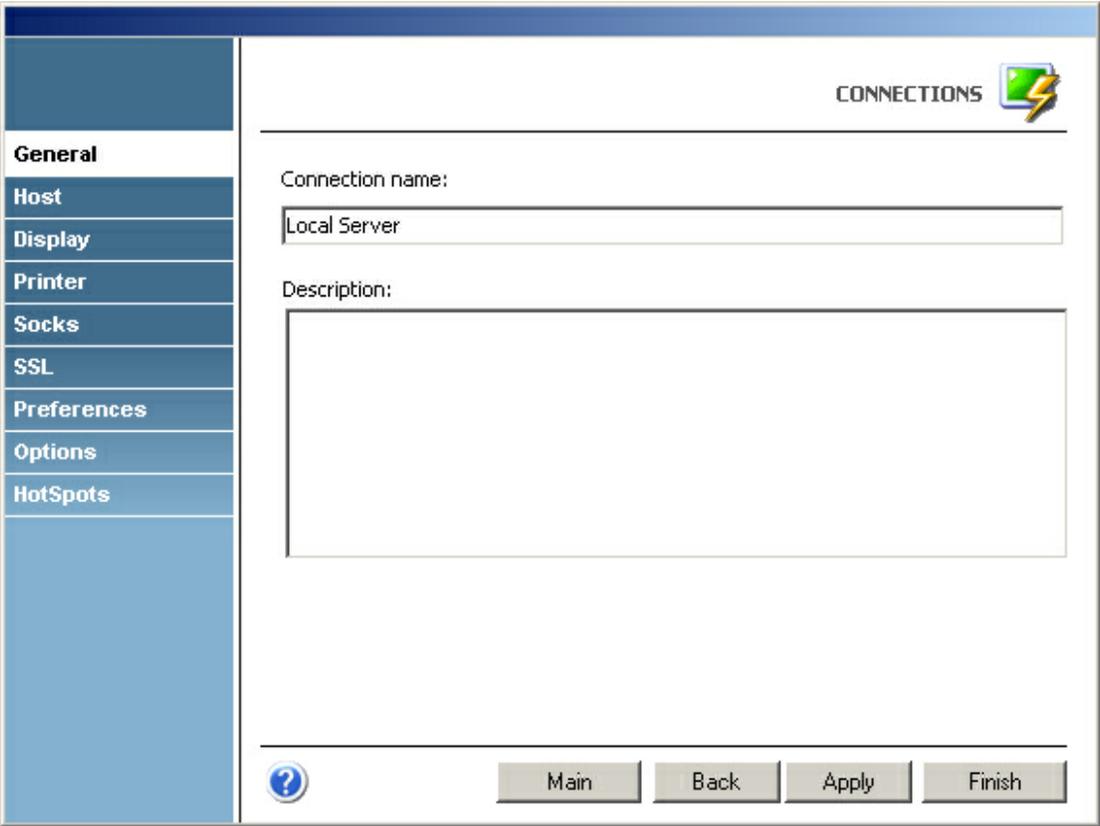
For a detailed description of the parameters available when editing Unix/VT connections, consult the following sections:

- [General](#)
- [Host](#)
- [Backup](#)
- [Display](#)

- [Printer](#)
- [Socks](#)
- [SSL](#)
- [SSH](#)
- [Gateway](#)
- [Preferences](#)
- [Options](#)
- [Hotspots](#)

4.1.1.1 General

In the 'General' tab you will find the following parameters:



The screenshot shows a software window titled "CONNECTIONS" with a lightning bolt icon. On the left is a vertical navigation menu with the following items: General (highlighted), Host, Display, Printer, Socks, SSL, Preferences, Options, and HotSpots. The main area contains the following fields:

- Connection name:** A text input field containing "Local Server".
- Description:** A large, empty text area for entering a description.

At the bottom of the window, there is a help icon (question mark in a circle) and four buttons: "Main", "Back", "Apply", and "Finish".

Connection name

In this field you must enter a name to identify the connection.

Description

Here you can type a longer description for the connection.

4.1.1.2 Host

In the 'Host' tab you will find the following parameters:

The screenshot shows the 'Host' configuration dialog box in the zScope Terminal Emulator. The dialog has a sidebar on the left with tabs: General, Host (selected), Display, Printer, Preferences, Options, and HotSpots. The main area is titled 'CONNECTIONS' and contains the following fields and options:

- Address:** Text box containing '200.168.0.43'
- Port:** Text box containing '23'
- Enable Keep Alive
- Socks Firewall
- Disable Telnet Protocol Negotiation
- Has Backup
- SSL
- SSH
- Gateway
- Character set translation:** Dropdown menu showing 'MsDos USA'

At the bottom of the dialog are buttons for '?', 'Main', 'Back', 'Apply', and 'Finish'.

Address

URL or IP address of the host computer.

Port

TCP port defined at the host computer for Telnet access.

- ⓘ Default Telnet port number is 23.

Enable Keep alive

Enables keep-alive mechanism, needed for some Telnet servers to prevent disconnections.

Socks Firewall

Enables support for Socks firewall.

- ⓘ When you check this option, the ['Socks' tab](#) will automatically appear in the connection settings dialog.

Disable Telnet Protocol Negotiation

Check this option if you want to omit the protocol negotiation when connecting.

Has Backup

Check this option if you would like to specify an alternate ip for this connection.

- ④ When you check this option, the ['Backup' tab](#) will automatically appear in the connection settings dialog.

SSL

Enables the SSL (Secure Sockets Layer) protocol for the host.

- ④ When you check this option, the ['SSL' tab](#) will automatically appear in the connection settings dialog. See [Using SSL](#). The SSL and Gateway options are mutually exclusive.

SSH

Enables the SSH protocol for the host.

- ④ When you check this option, the ['SSH' tab](#) will automatically appear in the connection settings dialog. The SSH and Gateway options are mutually exclusive.

Gateway

Check this option if there is a gateway for accessing the host.

- ④ When you check this option, the ['Gateway' tab](#) will automatically appear in the connection settings dialog. The Gateway options is mutually exclusive with the SSH and SSL options.

Character Set Translation

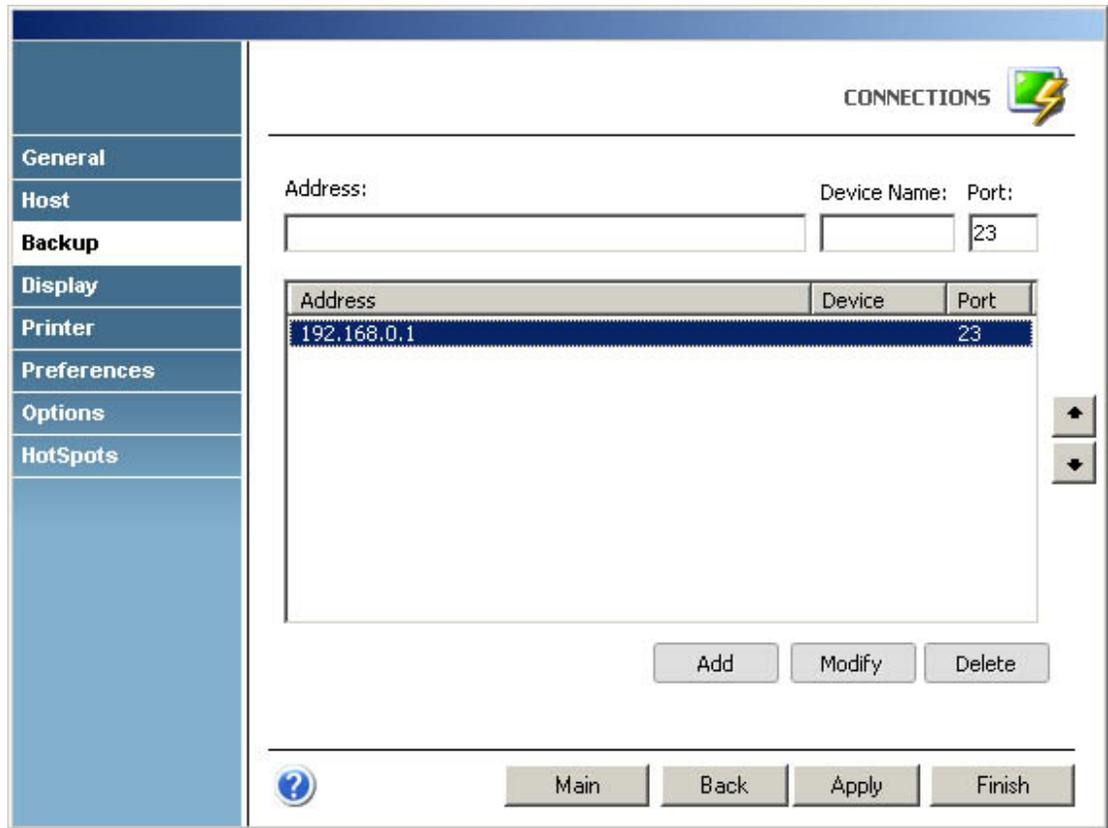
Select the character set that better suits your language needs.

- ④ When you select BiDi sets, a new option will be enabled so you can select the BiDi settings.

4.1.1.3 Backup

In the 'Backup' tab you will see a table showing the list of alternate hosts for the connection. If the connection fails, z/Scope will connect to an alternate host, starting with the first one in the list. If an alternate host fails, z/Scope will connect to the next one in the list.

To configure these settings you will find the following parameters:



Address

Enter here the IP address of the alternate host you would like to add for this connection

Device Name

Enter here the Device Name with which you would like to connect to this alternate host.

Device Name

Enter here the port number for this alternate host.



Move Up

Select a backup connection from the list and use this button to move it above other backup connections in the list.



Move Down

Select a backup connection from the list and use this button to move it below other backup connections in the list.

Add

Press this button to add the Address, Device Name and Port information entered above as a new host in the list. New hosts will be added last in the list.

Modify

Select a backup connection from the list and press this button to replace the

selected host with the information entered in the fields 'Address', 'Device Name' and 'Port'.

Delete

Select a backup connection from the list and press this button to delete it from the list.

4.1.1.4 Display

In the 'Display' tab you will find the following parameters:

The screenshot shows the 'CONNECTIONS' application window with the 'Display' tab selected in the left-hand menu. The main area is divided into three sections: 'Terminal', 'Screen Size', and 'Scrolling'. The 'Terminal' section has a 'Type' dropdown menu set to 'ANSI', a 'String' text box containing 'ANSI', a checked 'Automatic' checkbox, and an unchecked 'Auto wrap' checkbox. The 'Screen Size' section has a dropdown menu set to '24 rows x 80 cols', an unchecked 'Custom' checkbox, and input fields for 'Rows' (24) and 'Cols' (80). The 'Scrolling' section has a selected 'Smooth' radio button, an unselected 'Jump' radio button, and a 'Jump speed' input field set to 2. At the bottom of the window, there are buttons for 'Main', 'Back', 'Apply', and 'Finish'.

Terminal

Type / String

Specify the type of terminal to emulate, which is not necessarily the same that is informed to the server. To inform the server a different type of terminal than the one emulated, use the 'String' field. To automatically detect the type of terminal, check the 'Automatic' option.

DEC Answerback

Here you can specify the DEC "Transmit answerback message" control character. Check the 'Use Computer Name' Checkbox to assign the computer's name to the DEC Answerback field.

Auto Wrap

Check this option if you want the text lines to be wrapped when the terminal is resized.

Screen Size

Rows/Cols

Specify the number of rows and columns to be displayed.

Fixed Column Size

Check this option to display a horizontal scrollbar instead of resizing the font.

Scrollback lines

Specify the number of rows to keep in the buffer so they can be scrolled with the vertical scrollbar.

Scrolling

Smooth/Jump

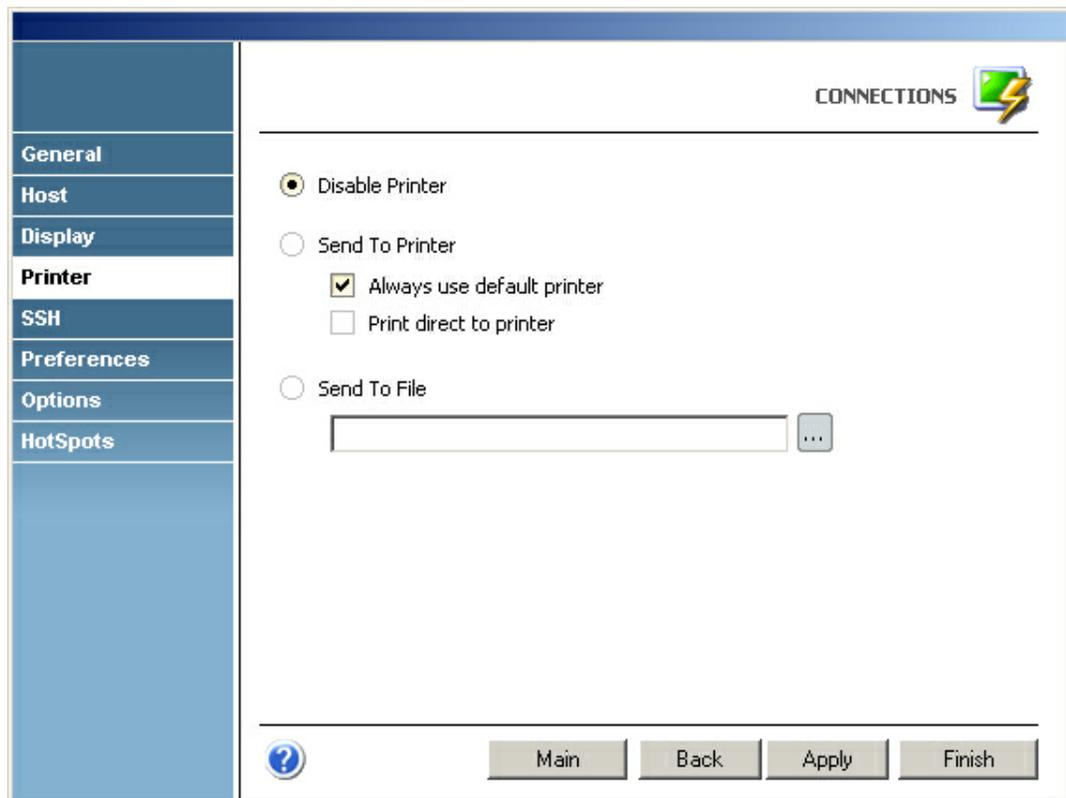
Select a method for scrolling.

Jump speed

Specify the number of rows to be scrolled when the scrolling method is set to 'Jump'.

4.1.1.5 Printer

In the 'Printer' tab you will find the following parameters:



Disable Printer

Select this option if you want to disable printing for this connection.

Send to Printer

Select this option to redirect the print output to the printer.

Always use default printer

Check this option if you want to use the default printer without prompting.

Print direct to printer

Check this option if you don't want to use the printing spool.

Send To File

Select this option to redirect the print output to a file.

4.1.1.6 Socks

In the 'Socks' tab you will find the following parameters:

The screenshot shows the 'CONNECTIONS' dialog box in the zScope Terminal Emulator. The left sidebar contains a list of tabs: General, Host, Display, Printer, **Socks**, SSH, Preferences, Options, and HotSpots. The 'Socks' tab is currently selected. The main content area is titled 'CONNECTIONS' and features a lightning bolt icon. It includes a 'Type' dropdown menu with 'svSocks4' selected, an 'Address' text input field, a 'Port' text input field, a checkbox for 'Requires authentication', a 'UserId' text input field, and a 'Password' text input field. At the bottom of the dialog are four buttons: 'Main', 'Back', 'Apply', and 'Finish'.

▲ This tab only becomes available when the 'Socks' option is checked in the ['Host' tab](#).

Type

Indicates the type of Socks protocol you will be connecting to. z/Scope provides support for svSocks 4, 4A and 5 protocols.

Address

In this field you must enter the IP address of the Socks server.

Port

In this field you must enter the port number of the Socks service at the host.

Requires Authentication

When connecting with svSocks4A protocol and higher, you have the option of providing a User ID and a Password for authentication.

UserId

In this field you must enter your User ID.

Password

In this field you must enter your Password.

4.1.1.7 SSL

In the 'SSL' tab you will find the following parameters:

The screenshot shows the 'CONNECTIONS' settings window with the 'SSL' tab selected. The 'SSL Method' section has 'SSL 2/3' selected. The 'Server Certificate' section has 'Accept invalid CA certificates' checked. The 'Client Certificate' section has empty text boxes for 'Root CA Cert. File', 'Client Cert. File', 'Key File', and 'Cert. Password'.

 This tab only becomes available when the 'SSL' option is checked in the ['Host' tab](#).

SSL Method

Choose one of the available methods shown: SSL 2.0, SSL 3.0 or TLS 1.0.

Server Certificate

The 'Display certificate' option controls whether to show the Certificate Info immediately after establishing the connection. The other options refer to the policy adopted when dealing with certificates that do not meet certain security conditions.

Client Certificate

Enter the file name of the certificate files that you own.

4.1.1.8 SSH

In the 'SSH' tab you will find the following parameters:

The screenshot shows the 'SSH' configuration window. The left sidebar is a vertical menu with the following items: General, Host, Display, Printer, Socks, SSH (highlighted), Preferences, Options, and HotSpots. The main content area is titled 'CONNECTIONS' and contains the following settings:

- Protocol Version:** Three radio buttons are present: 'SSH 1 Only' (unselected), 'SSH 1/2' (selected), and 'SSH 2' (unselected).
- Enable Compression:** A checked checkbox.
- Authentication:** A section containing:
 - Username:** A text field with the value 'XXXXXXXXXX'.
 - Password:** An empty text field.
 - Private Key File for Authentication:** An unchecked checkbox.
 - Key File:** A text field with a browse button (three dots) to its right.

At the bottom of the window, there is a help icon (question mark) and four buttons: 'Main', 'Back', 'Apply', and 'Finish'.

⚠ This tab only becomes available when the 'SSH' option is checked in the ['Host' tab](#).

SSH Protocol Version

Choose one of the available versions: SSH 1 Only, SSH 1/2 or SSH 2 Only.

Enable Compression

Check this option to enable compression for the SSH protocol.

Authentication

Username

Enter an user name with access to the host via the SSH protocol.

Password

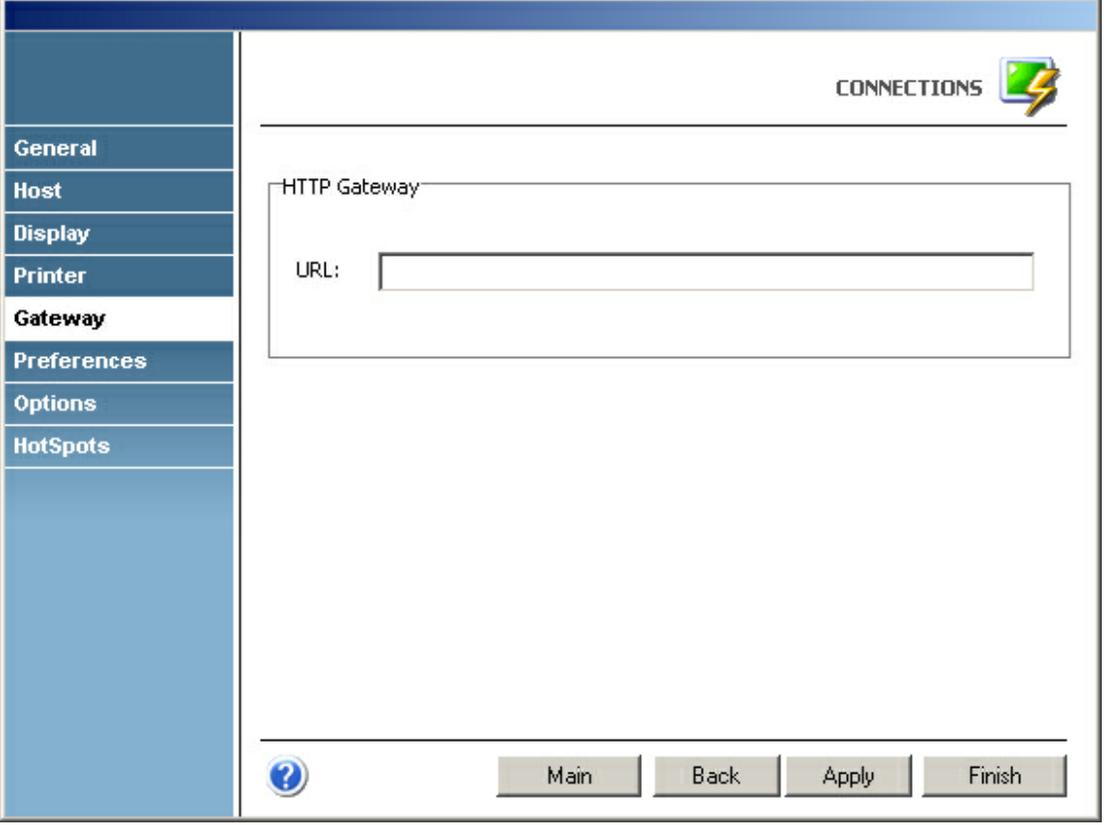
Enter the password for the specified user name.

Private Key File for Authentication

Check this option if you want to use a private key-file for the authentication process. You must enter the path of the file in the field below.

4.1.1.9 Gateway

In the 'Gateway' tab you will find the following parameters:

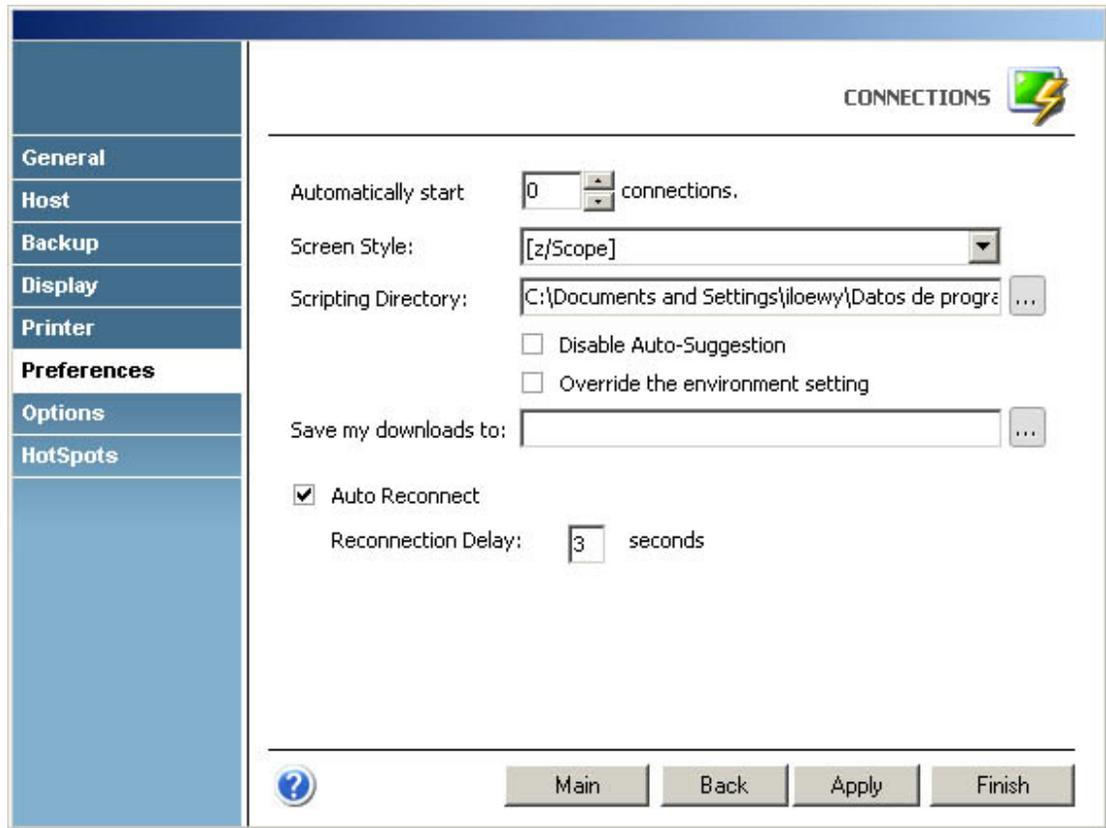


The screenshot shows a software window titled "CONNECTIONS" with a lightning bolt icon. On the left is a vertical menu with the following items: General, Host, Display, Printer, Gateway (highlighted), Preferences, Options, and HotSpots. The main area is titled "HTTP Gateway" and contains a "URL:" label followed by a text input field. At the bottom of the window, there is a help icon (a question mark in a circle) and four buttons: "Main", "Back", "Apply", and "Finish".

URL
Enter here the gateway's URL.

4.1.1.10 Preferences

In the 'Preferences' tab you will find the following parameters:



Automatically Start [n] Connections

Allows you to specify the number of sessions of this connection that will be automatically established upon z/Scope start.

Screen Style

Allows you to select a default [Screen Style](#) for this connection.

Scripting Directory

Specify the folder on your local computer where script files will be stored.

Disable Auto-Suggestion

Check this option to have the [Auto-Suggestion](#) feature disabled by default for this connection.

Override the Environment Setting

Check this option to override environment settings with the connection's settings.

Auto Reconnect

Check this option if you would like to automatically reconnect to the host after logging off.

Reconnection Delay

Specify in this field the amount of time (in seconds) that you would like the system to take before auto reconnecting to the host.

4.1.1.11 Options

In the 'Options' tab you will find the following parameters:

The screenshot shows the 'Options' tab in the CONNECTIONS settings window. The sidebar on the left contains the following tabs: General, Host, Display, Printer, SSH, Preferences, Options (selected), and HotSpots. The main content area is titled 'CONNECTIONS' and contains the following settings:

- Modes:**
 - Local echo
 - Auto repeat
 - Break enabled
- Receive Replacements:**
 - CR is: CR LF CR/LF
 - LF is: CR LF CR/LF
- Send Replacements:**
 - Enter sends: CR CR/LF
 - Backspace sends: Delete Backspace
 - Cursor keys: Normal Application
 - Keypad: Numeric Application
- LineMode:**
 - Mode:

At the bottom of the window, there is a help icon (question mark) and four buttons: Main, Back, Apply, and Finish.

Modes

Local Echo

Check this option to allow local echoing of the characters when the server does not return echoes.

Auto repeat

Check this option to enable the auto repeat feature for the keyboard.

Break enabled

Check this option to be able to use the break command.

Receive Replacements

CR/LF is

Select the desired behaviour for the 'Carriage Return' (CR) and 'Line Feed' (LF) commands.

Send Replacements

Enter/Backspace sends

Select the desired behaviour for the 'Enter' and 'Backspace' keys.

Cursor/Keypad keys

Specify how the cursor and keypad keys are interpreted.

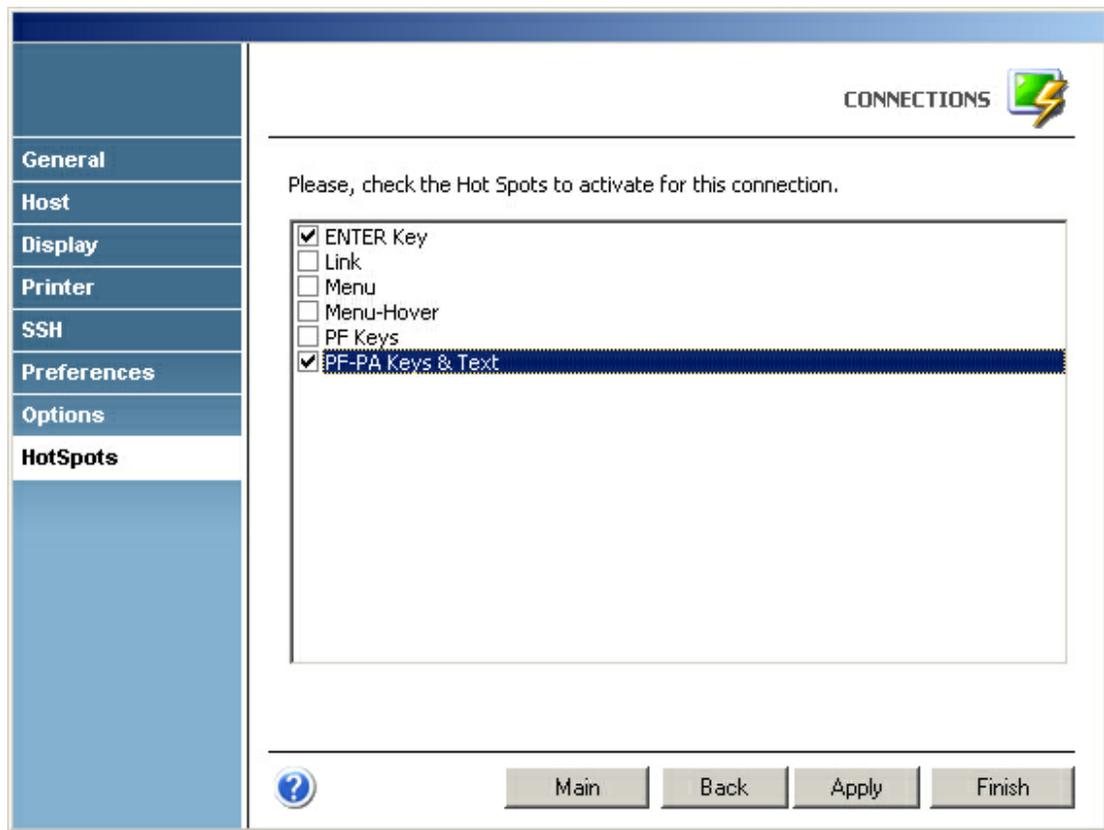
Line Mode

Mode

Indicate when LineMode will be activated from the options available in the combobox.

4.1.1.12 HotSpots

In the 'HotSpots' tab you can choose the HotSpots that will be available when working with the connection.



Related Topics

- [Hotspots Settings](#)

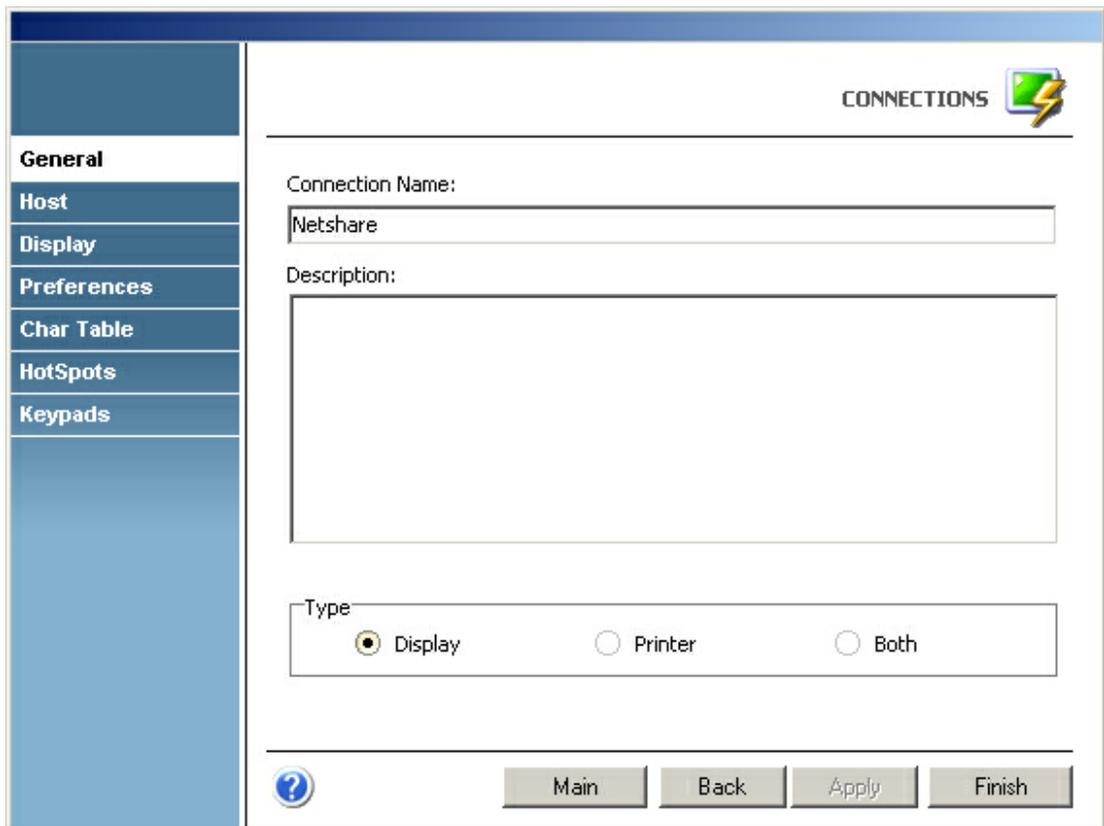
4.1.2 Mainframe / AS400 Connections

For a detailed description of the parameters available when editing Mainframe/AS400 connections, consult the following sections:

- [General](#)
- [Host](#)
- [Backup](#)
- [Display](#)
- [Printer](#)
- [Socks](#)
- [SSL](#)
- [Gateway](#)
- [Preferences](#)
- [Char Table](#)
- [Hotspots](#)
- [Keypads](#)

4.1.2.1 General

In the 'General' tab you will find the following parameters:



The screenshot shows a software dialog box titled "CONNECTIONS" with a lightning bolt icon. On the left is a vertical menu with the following items: "General" (highlighted), "Host", "Display", "Preferences", "Char Table", "HotSpots", and "Keypads". The main area contains the following fields and controls:

- Connection Name:** A text input field containing "Netshare".
- Description:** A large empty text area.
- Type:** A group box containing three radio buttons: "Display" (selected), "Printer", and "Both".
- Buttons:** A row of four buttons: "Main" (with a question mark icon), "Back", "Apply", and "Finish".

Connection Name

In this field you must enter a name for the connection.

Description

Here you can type a description for the connection.

Type

Here you can specify if the connection has a printer session associated to it or if it is a standalone printer or display session.

- ④ When you select the 'Printer' or the 'Both' options, the ['Printer' tab](#) will automatically appear in the connection settings dialog.

4.1.2.2 Host

In the 'Host' tab you will find the following parameters:

The screenshot shows a configuration window for a host connection. The sidebar on the left lists various configuration categories, with 'Host' currently selected. The main configuration area includes an 'Address' field containing 'netshare400.com' and a 'Port' field containing '23'. There are several checkboxes: 'Extended' is checked, while 'Enable Keep Alive', 'Has Backup', 'SSL', 'Socks Firewall', and 'Gateway' are unchecked. A section titled 'TN5250E Extended Info' contains two input fields for 'User Id' and 'Password'. At the bottom of the window, there are buttons for '?', 'Main', 'Back', 'Apply', and 'Finish'.

Address

URL or IP address of the host computer.

Port

TCP port defined at the host computer for Telnet access.

- ④ Default Telnet port number is 23.

Extended

Enables Telnet Extended protocols (TN3270E or TN5250E). This enables 'User Id' and 'Password' input boxes for AS/400 connections.

Enable Keep alive

Enables keep-alive mechanism, needed for some Telnet servers.

Has Backup

Check this option if you would like to specify an alternate ip for this connection.

- ④ When you check this option, the '[Backup](#)' tab will automatically appear in the

[connection settings dialog.](#)

SSL

Enables the SSL (Secure Sockets Layer) protocol for the host.

- ④ When you check this option, the 'SSL' tab will automatically appear in the connection settings dialog. See '[SSL' Tab](#). The SSL and Gateway options are mutually exclusive.

Socks Firewall

Enables support for Socks firewall.

- ④ When you check this option, the 'Socks' tab will automatically appear in the connection settings dialog. See '[Socks' tab](#).

Gateway

Check this option if there is a gateway for accessing the host.

- ④ When you check this option, the 'Gateway' tab will automatically appear in the connection settings dialog. See '[Gateway' tab](#). The SSL and Gateway options are mutually exclusive.

TN5250E Extended Info

These options, only available for TN5250 connections provide the user with an enhanced security method defined by the TN5250E norm.

4.1.2.3 Backup

In the 'Backup' tab you will see a table showing the list of alternate hosts for the connection. If the connection fails, z/Scope will connect to an alternate host, starting with the first one in the list. If an alternate host fails, z/Scope will connect to the next one in the list.

To configure these settings you will find the following parameters:

The screenshot shows a software interface for managing connections. On the left is a vertical sidebar with a blue background and white text for menu items: General, Host, Backup, Display, Printer, Preferences, Char Table, HotSpots, and Keypads. The main content area has a white background and a blue header bar with the title 'CONNECTIONS' and a lightning bolt icon. Below the header, there are three input fields: 'Address:', 'Device Name:', and 'Port:'. The 'Port:' field contains the number '23'. Underneath these fields is a table with three columns: 'Address', 'Device', and 'Port'. The table is currently empty. To the right of the table are two small square buttons with up and down arrows. Below the table are three buttons: 'Add', 'Modify', and 'Delete'. At the bottom of the window, there is a blue question mark icon and four buttons: 'Main', 'Back', 'Apply', and 'Finish'.

Address

Enter here the IP address of the alternate host you would like to add for this connection

Device Name

Enter here the Device Name with which you would like to connect to this alternate host.

Port

Enter here the port number for this alternate host.



Move Up

Select a backup connection from the list and use this button to move it above other backup connections in the list.



Move Down

Select a backup connection from the list and use this button to move it below other backup connections in the list.

Add

Press this button to add the Address, Device Name and Port information entered above as a new host in the list. New hosts will be added last in the list.

Modify

Select a backup connection from the list and press this button to replace the

selected host with the information entered in the fields 'Address', 'Device Name' and 'Port'.

Delete

Select a backup connection from the list and press this button to delete it from the list.

4.1.2.4 Display

In the 'Display' tab you will find the following parameters:

The screenshot shows the 'CONNECTIONS' dialog box with the 'Display' tab selected. The left sidebar contains a list of tabs: General, Host, Display (selected), Preferences, Char Table, HotSpots, and Keypads. The main area contains the following settings:

- Display Type:** Radio buttons for '80 Columns' (selected) and '132 Columns'.
- Extended Attributes:** A checked checkbox.
- SYSREQ Command Dialog:** An unchecked checkbox.
- Device Name:** An empty text input field.
- Device Name Suffix:** Radio buttons for 'None' (selected), 'Letter (A-Z)', and 'Number (1-9)'.

At the bottom, there is a help icon (question mark) and four buttons: 'Main', 'Back', 'Apply', and 'Finish'.

Display Type

Select the desired resolution for the host.

- Available resolutions vary according to the type of host you are connecting to.

Extended Attributes

Enables extended attributes for the connection.

SYSREQ Command Dialog

Only for AS/400 connections: enables a command line for the SysReq function.

Device Name

Specifies the logical unit or device name for the connection.

Device Name Suffix

Allows you to specify a suffix method to use for several connections.

4.1.2.5 Printer

The printer tabs for AS/400 and mainframe connections differ and they will be described in two different chapters:

- [AS/400 Printer Tab](#)
- [Mainframe Printer Tab](#)

4.1.2.5.1 AS/400

In the 'Printer' tab for an AS/400 connection you will find the following parameters:

The screenshot shows the 'CONNECTIONS' dialog box with the 'Printer' tab selected. The left sidebar contains the following tabs: General, Host, Display, Printer (selected), Preferences, Char Table, HotSpots, and Keypads. The main area contains the following fields and controls:

- Device Name: [Text Field]
- Connect on z/Scope start
- Print to folder [Text Field] [...]
- Message Queue: [QSYSOPR] Host Font: [Courier 10]
- Message Library: [*LIBL] Forms Mode: [Autocut]
- Host Print Transform
- Model: [*NONE] Paper Source 1: [*NONE] Paper Source 2: [*NONE] Envelope Source: [*NONE]
- Workstation Customizing Object: Name: [Text Field] Library: [Text Field]
- ASCII Code Page 889

At the bottom of the dialog are buttons for Main, Back, Apply, and Finish, and a help icon.

- ▲ This tab only becomes available when either the 'Printer' or the 'Both' options are checked in the ['General' tab](#).

Device Name

In this field you must enter the host device name.

Connect on z/Scope Start

Check this option if you want the printer session to be started automatically when z/Scope starts independently of the display session.

Print to folder

Check this option to redirect the output to the specified folder.



The following parameters only apply to 5270 Connections.

Message Queue

In this field you must enter the host's queue name.

Message Library

In this field you must enter the host's library name.

Host Font

In this field you must enter the name of the host font.

Forms Mode

In this field you must select the desired forms mode.

Host Print Transform

Check this option to enable the host print transform function for the printer. This function is only to be used for ASCII printers and enables the following options:

Model

In this field you must enter the printer model.

Paper Source 1

In this field you must specify the size of the paper in drawer 1 or continuous size paper if applicable.

Paper Source 2

In this field you must specify the size of the paper in drawer 2. This value is ignored for continuous feed printers.

Envelope Source

In this field you must specify the size of the envelope.

Workstation Customizing Object

These parameter can be used by the host print transform function when a user-defined workstation customizing object is necessary. A user-defined customizing object for the printer is usually not needed due to the extended support provided by the host print transform function.

ASCII Code Page 889

Check this option if your printer supports code page 899. Code page 899 is not resident on most ASCII printers. With the IBM 4029 LaserPrinter, a font card is required.

4.1.2.5.2 Mainframe

In the 'Printer' tab for a mainframe connection you will find the following parameters:

The screenshot shows a software window titled 'CONNECTIONS' with a lightning bolt icon. On the left is a vertical navigation menu with the following items: General, Host, Display, Printer (highlighted), Preferences, Char Table, HotSpots, and Keypads. The main area contains the following settings:

- Device Name: [text input field] Associated
- Connect on z/Scope start
- Print to folder [text input field] [three dots menu icon]

At the bottom of the window are four buttons: Main, Back, Apply, and Finish, along with a help icon (question mark in a circle).

▲ This tab only becomes available when either the 'Printer' or the 'Both' options are checked in the ['General' tab](#).

Device Name

In this field you must enter the host device name.

Associated

Check this option if you want the printer session to start whenever the display session is started.

Connect on z/Scope Start

Check this option if you want the printer session to be started automatically when z/Scope starts independently of the display session..

Print to folder

Check this option to redirect the output to the specified folder.

4.1.2.6 Socks

In the 'Socks' tab you will find the following parameters:

The screenshot shows the 'CONNECTIONS' dialog box with the 'Socks' tab selected. The sidebar on the left lists various configuration categories. The main panel contains the following fields and controls:

- Type:** A dropdown menu currently set to 'svSocks4'.
- Address:** A text input field.
- Port:** A text input field.
- Requires Authentication:** An unchecked checkbox.
- UserId:** A text input field.
- Password:** A text input field.

At the bottom of the dialog, there is a help icon (question mark in a circle) and four buttons: 'Main', 'Back', 'Apply', and 'Finish'.

 This tab only becomes available when the 'Socks' option is checked in the ['Host' tab](#).

Type

Indicates the type of Socks protocol you will be connecting to. z/Scope provides support for svSocks 4, 4A and 5 protocols.

Address

In this field you must enter the IP address of the Socks server.

Port

In this field you must enter the port number of the Socks service at the host.

Requires Authentication

When connecting with svSocks4A protocol and higher, you have the option of providing a User ID and a Password for authentication.

UserId

In this field you must enter your User ID.

Password

In this field you must enter your Password.

4.1.2.7 SSL

In the 'SSL' tab you will find the following parameters:

The screenshot shows a software window titled 'CONNECTIONS' with a lightning bolt icon. On the left is a vertical navigation menu with the following items: General, Host, Display, Printer, SSL (highlighted), Preferences, Char Table, HotSpots, and Keypads. The main area contains the following settings:

- SSL Method:** Radio buttons for SSL 2/3 (selected), SSL 2.0, SSL 3.0, and TLS 1.0.
- Server Certificate:** Checkboxes for Display Certificate, Accept Any Invalid Certificate, Accept Expired Certificates, Accept Certificates Not Yet Valid, Accept Invalid CA Certificates (checked), and Accept Self Signed Certificates.
- Client Certificate:** Four text input fields with browse buttons (three dots): Root CA Cert. File, Client Cert. File, Key File, and Cert. Password.

At the bottom of the window are buttons for Main, Back, Apply, and Finish, along with a help icon (question mark in a circle).

▲ This tab only becomes available when the 'SSL' option is checked in the ['Host' tab](#).

SSL Method

Choose one of the available methods shown: SSL 2.0, SSL 3.0 or TLS 1.0.

Server Certificate

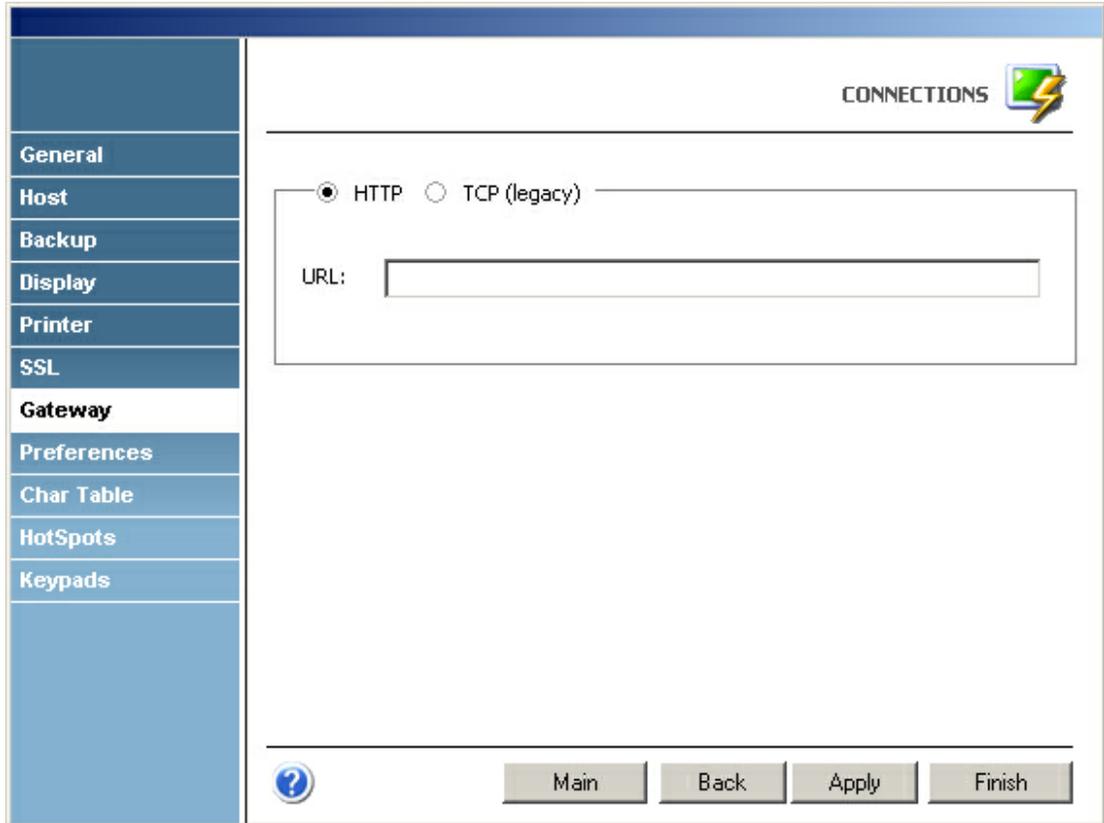
The 'Display certificate' option controls whether to show the Certificate Info immediately after establishing the connection. The other options refer to the policy adopted when dealing with certificates that do not meet certain security conditions.

Client Certificate

Enter the file name of the certificate files that you own.

4.1.2.8 Gateway

In the 'Gateway' tab you will find the following parameters:



The screenshot shows the 'Gateway' configuration window in the zScope Terminal Emulator. The window has a title bar with 'CONNECTIONS' and a lightning bolt icon. On the left is a vertical navigation pane with the following tabs: General, Host, Backup, Display, Printer, SSL, Gateway (selected), Preferences, Char Table, HotSpots, and Keypads. The main area contains two radio buttons: 'HTTP' (selected) and 'TCP (legacy)'. Below these is a text input field labeled 'URL:'. At the bottom of the window are four buttons: a help button (question mark icon), 'Main', 'Back', 'Apply', and 'Finish'.

HTTP/TCP

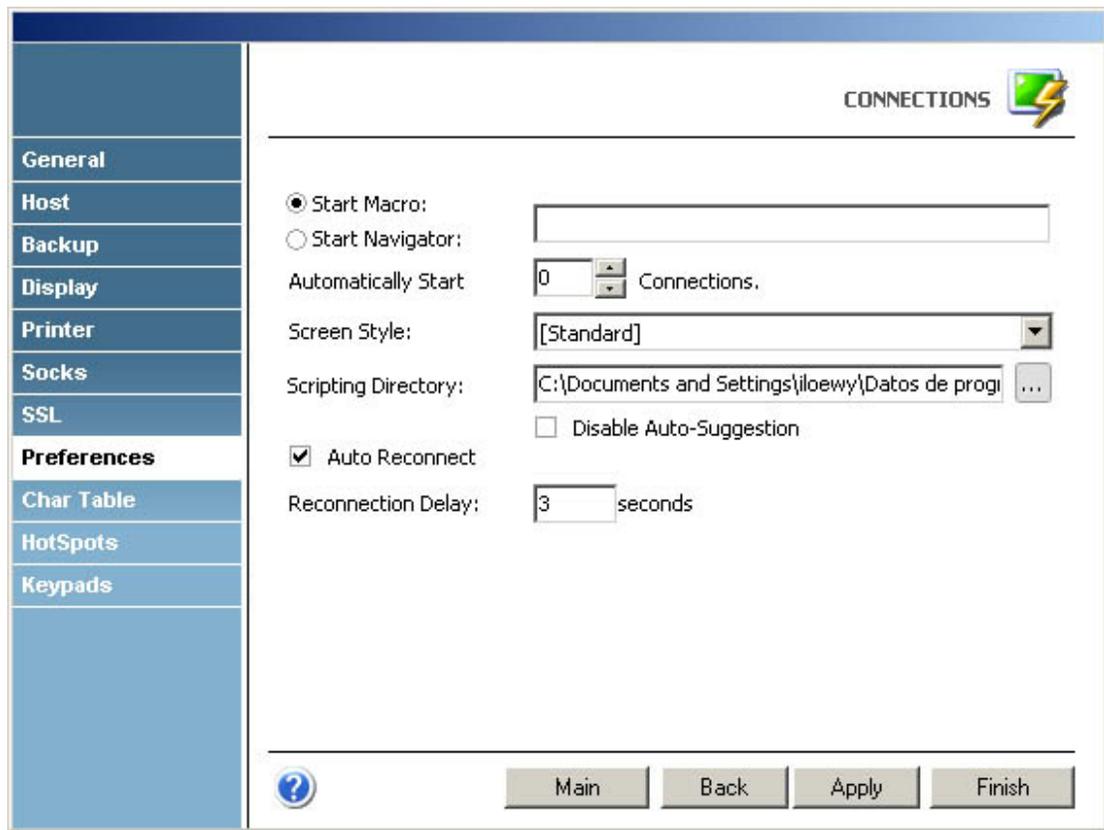
Choose the protocol that the gateway uses for accessing the host.

URL

Enter here the gateway's URL.

4.1.2.9 Preferences

In the 'Preferences' tab you will find the following parameters:



Start Macro/Navigator

Select a [Macro](#) or a [Navigator](#) to start automatically when connecting.

Automatically Start [n] Connections

Allows you to specify the number of sessions of this connection that will be automatically established upon z/Scope start.

Screen Style

Allows you to select a default [Screen Style](#) for this connection.

Scripting Directory

Specify the folder on your local computer where script files will be stored.

Disable Auto-Suggestion

Check this option to have the [Auto-Suggestion](#) feature disabled by default for this connection.

Auto Reconnect

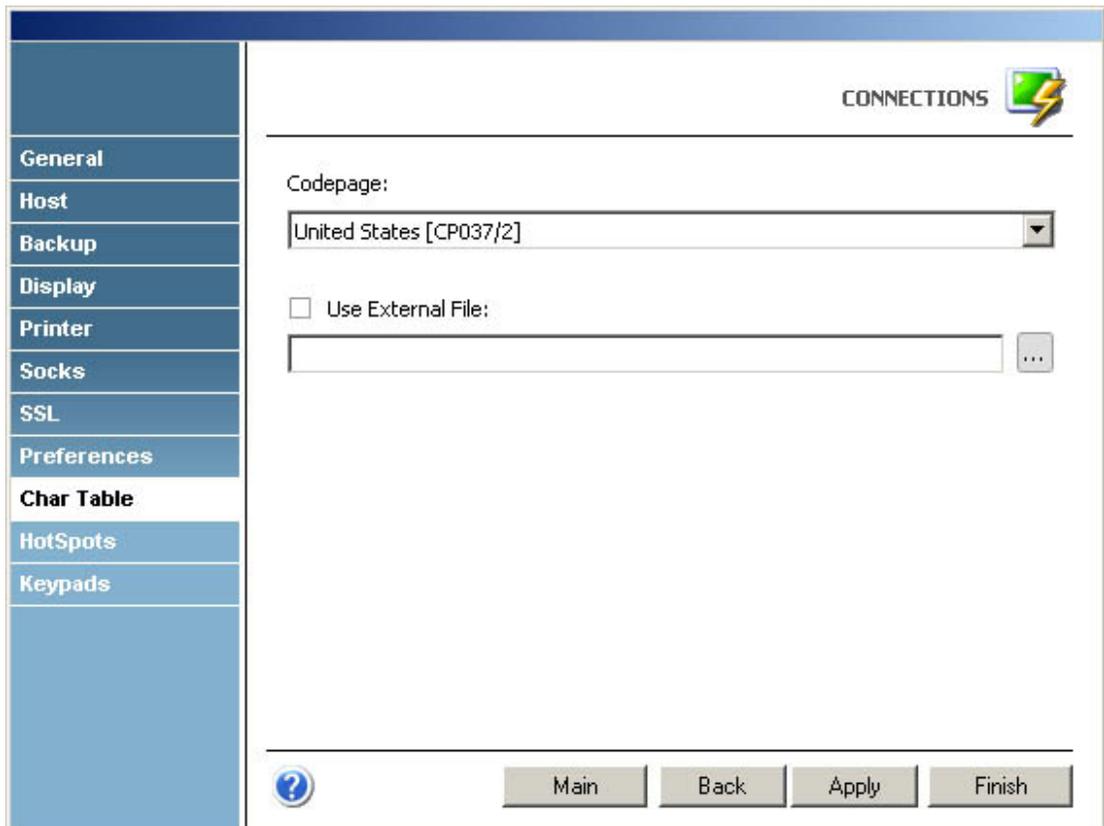
Check this option if you would like to automatically reconnect to the host after logging off.

Reconnection Delay

Specify in this field the amount of time (in seconds) that you would like the system to take before auto reconnecting to the host.

4.1.2.10 Char Table

In the 'Char Table' tab you will find the following parameters:



Codepage

Allows you to select an internal Character Conversion Table. See [Internal Conversion Tables](#).

Use External File

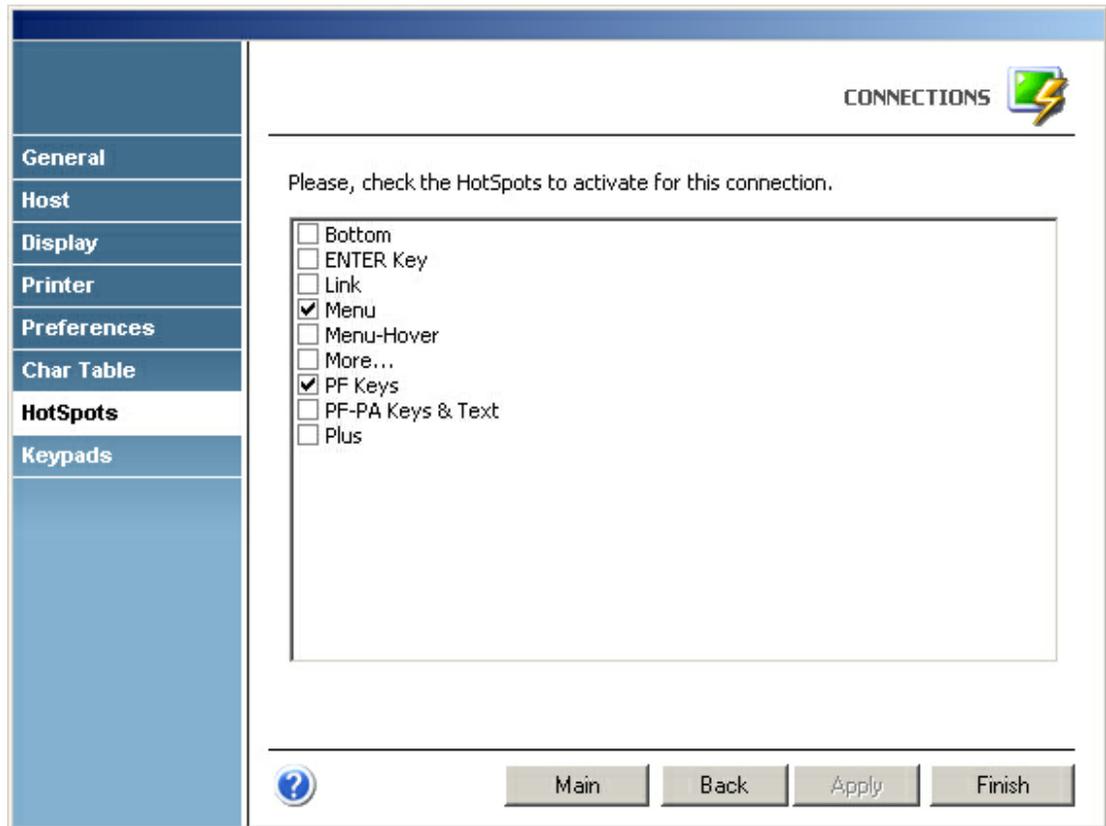
Mark this option if you to additionally enter the file name (.ebc) of an external Character Conversion Table to be used for the connection.

Related Topics

- [Internal Conversion Tables](#)
- [Modifying the Character Table](#)
- [Using an external Character Table](#)

4.1.2.11 HotSpots

In the 'HotSpots' tab you can choose the HotSpots that will be available when working with the connection.

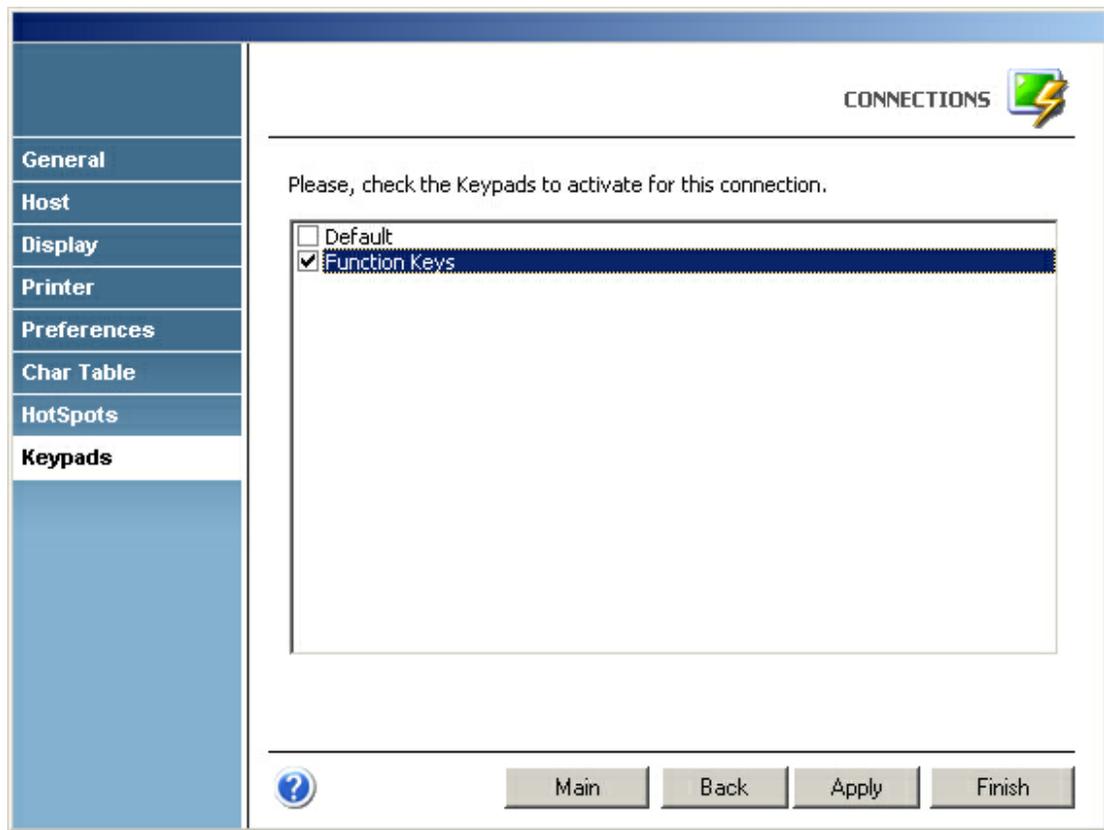


Related Topics

- [HotSpots Settings](#)
- [Using HotSpots](#)

4.1.2.12 Keypads

In the 'Keypads' tab you can choose the Keypads that will be available when working with the connection.



Related Topics

- Keypads Settings

4.1.3 FTP

For a detailed description of the parameters available when editing FTP connections, consult the following sections:

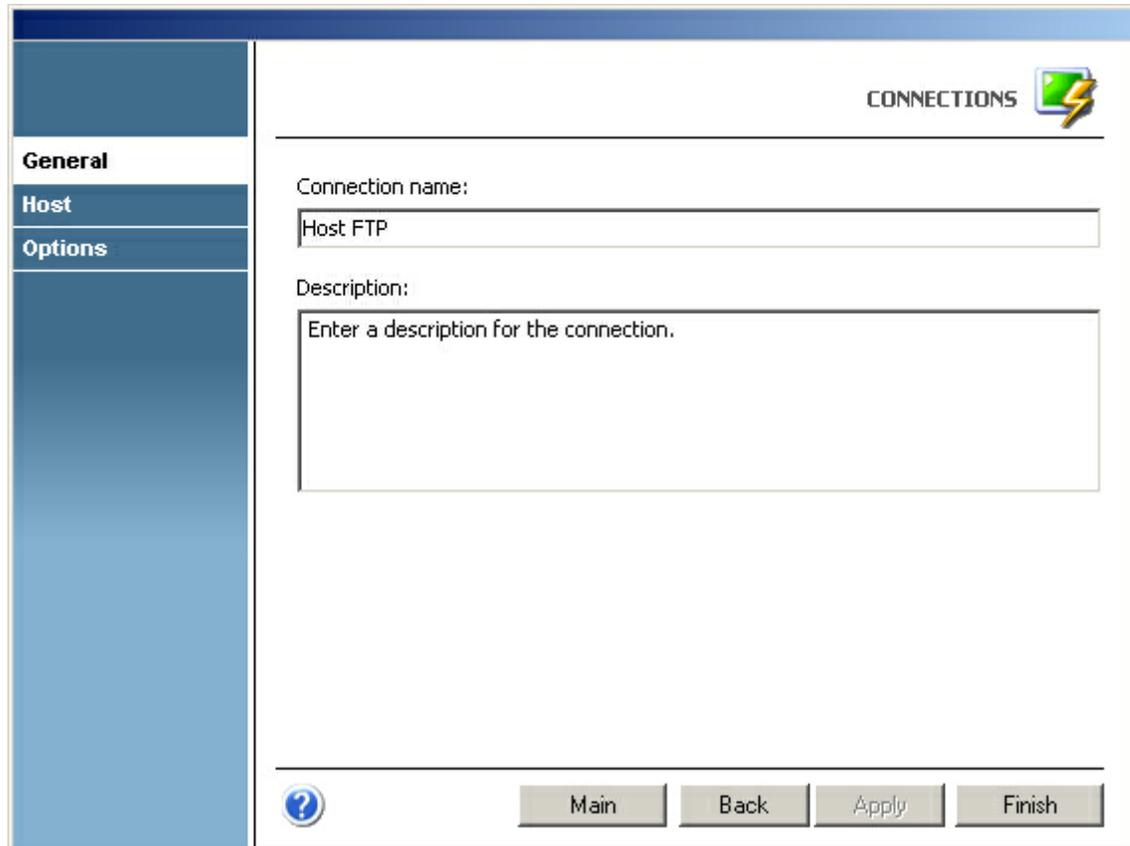
- [General](#)
- [Host](#)
- [FTP Options](#)
- [SSL](#)

Related Topics

Setting up a Connection

4.1.3.1 General

In the 'General' tab you will find the following parameters:



The screenshot shows a software window titled 'CONNECTIONS' with a lightning bolt icon. On the left is a navigation pane with three tabs: 'General' (selected), 'Host', and 'Options'. The main area contains two input fields: 'Connection name:' with the text 'Host FTP' and 'Description:' with the placeholder text 'Enter a description for the connection.'. At the bottom, there is a help icon (question mark in a circle) and four buttons: 'Main', 'Back', 'Apply', and 'Finish'.

Connection name

In this field you must enter a name the FTP connection.

Description

In this field you can type a description for the FTP connection.

4.1.3.2 Host

In the 'Host' tab you will find the following parameters:

The screenshot shows the 'CONNECTIONS' dialog box in the zScope Terminal Emulator. The 'Host' tab is selected in the sidebar. The main area contains the following fields and options:

- Address:** 192.168.0.180
- Port:** 21
- SSL**
- UserId:** p390
- Password:** *****
- Anonymous Login**
- Show Login Dialog**

At the bottom of the dialog, there is a help icon (?) and four buttons: Main, Back, Apply, and Finish.

Address

URL or IP address of the host machine that will act as the FTP server.

Port

TCP port defined for FTP access at the remote computer.

 Default FTP port number is 21.
If you check the 'SSL' option, the default port number will change to 990.

SSL

Check this option if the host you are connecting to supports the SSL protocol. See [Security and Encryption](#).

 When you check this option, the 'SSL' tab will automatically appear in the connection settings dialog. See ['SSL' tab](#).

UserId

In this field you must enter your User ID.

Password

In this field you must enter your Password.

Anonymous Login

Enable this option if you prefer to log in anonymously.

Show Login Dialog

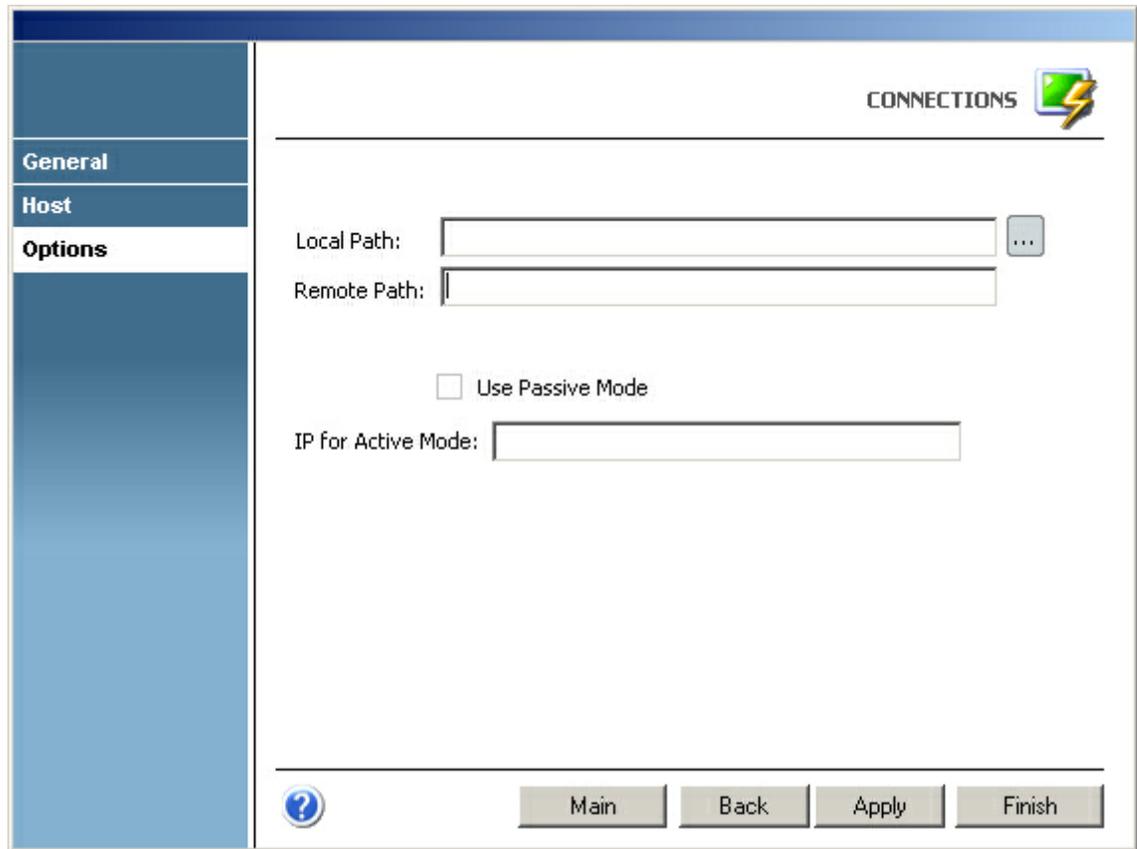
Check this option if you want the 'Login' dialog to be shown every time you connect to the remote computer.



Check the 'Don't ask again' option in the 'Login' dialog if you want z/Scope to remember your login information so that next time you connect you don't have to re-enter your User ID and Password.

4.1.3.3 Options

In the 'FTP Options' tab you will find the following parameters:



Local Path

Here you can set the default path in your local computer. When you start an FTP connection, the 'My Computer' panel will be positioned in the folder you specify in this field.

Remote Path

Here you can set the default path in the remote computer. When you start an FTP connection, the 'FTP Server' panel will be positioned in the folder you specify in this field.

Use Passive Mode

If this option is checked, a PASV command will be sent to tell the host it is working in passive mode.

IP for Active Mode

Use this field to specify the IP address you will be using if working in active mode.

4.1.3.4 SSL

In the 'SSL' tab you will find the following parameters:

The screenshot shows the 'CONNECTIONS' settings window. The 'SSL' tab is active in the sidebar. The main content area is titled 'CONNECTIONS' and contains the following options:

- SSL Method:** Radio buttons for 'Implicit SSL' (selected), 'Auth SSL', and 'Auth TLS'.
- Secure file listing:** A checkbox that is currently unchecked.
- Server Certificate:** A group of checkboxes:
 - Display certificate
 - Accept expired certificates
 - Accept invalid CA certificates
 - Accept any invalid certificate
 - Accept certificates not yet valid
 - Accept self signed certificates

At the bottom of the window, there are buttons for 'Main', 'Back', 'Apply', and 'Finish', along with a help icon (question mark in a circle).

 This tab only becomes available when the 'SSL' option is checked in the ['Host' tab](#).

SSL Method

Choose one of the available methods shown: Implicit SSL, Auth SSL or Auth TLS.

Secure File Listing

Check this option if you want to encrypt file listing data as well as file data during transfers.

Server Certificate

The 'Display Certificate' option controls whether to show the Certificate Info immediately after establishing the connection. The other options refer to the policy adopted when dealing with certificates that do not meet certain security conditions.

Related Topics

[Security and Encryption Using SSL](#)

4.1.4 Remote Desktop

For a detailed description of the parameters available when editing RD connections, consult the following sections:

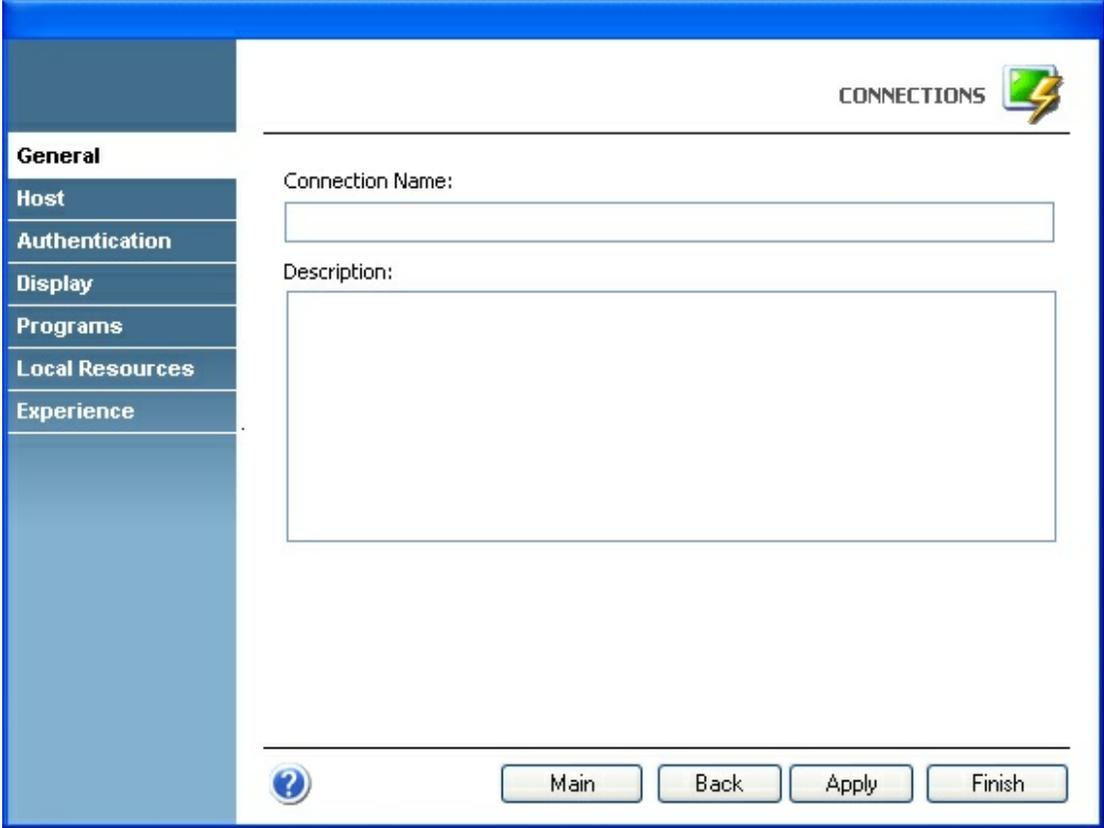
- [General](#)
- [Host](#)
- FTP Options
- SSL

Related Topics

Setting up a Connection

4.1.4.1 General

In the 'General' tab you will find the following parameters:



The screenshot shows the 'CONNECTIONS' dialog box with the 'General' tab selected. The dialog has a blue header with the title 'CONNECTIONS' and a lightning bolt icon. On the left is a vertical navigation pane with tabs for 'General', 'Host', 'Authentication', 'Display', 'Programs', 'Local Resources', and 'Experience'. The 'General' tab is active. The main area contains two input fields: 'Connection Name:' with a single-line text box, and 'Description:' with a larger multi-line text box. At the bottom, there is a help icon (question mark in a circle) and four buttons: 'Main', 'Back', 'Apply', and 'Finish'.

Connection name

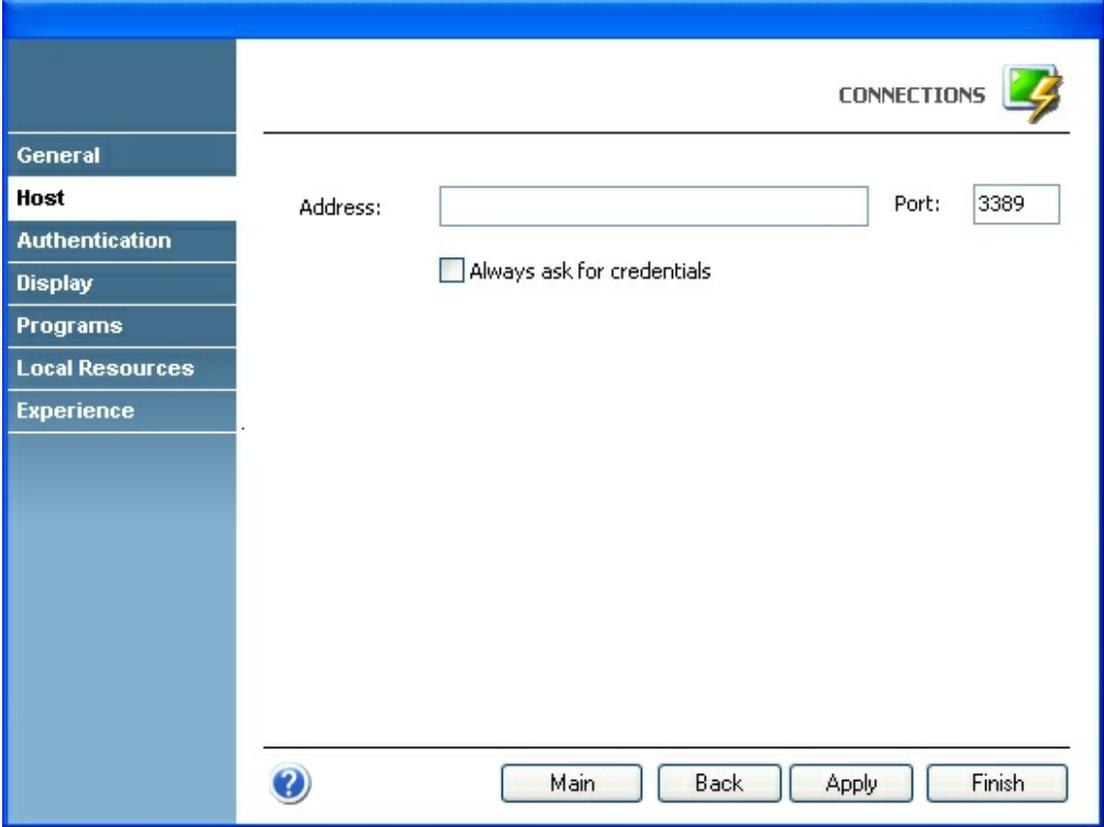
In this field you must enter a name the FTP connection.

Description

In this field you can type a description for the FTP connection.

4.1.4.2 Host

In the 'Host' tab you will find the following parameters:



The screenshot shows a software window titled 'CONNECTIONS' with a lightning bolt icon. On the left is a vertical navigation pane with tabs: 'General', 'Host' (selected), 'Authentication', 'Display', 'Programs', 'Local Resources', and 'Experience'. The main area contains the following fields and controls:

- 'Address:' followed by an empty text input box.
- 'Port:' followed by a text input box containing '3389'.
- A checkbox labeled 'Always ask for credentials' which is currently unchecked.
- At the bottom, there is a help icon (question mark in a circle) and four buttons: 'Main', 'Back', 'Apply', and 'Finish'.

Address

URL or IP address of the host machine that will act as the FTP server.

Port

TCP port defined for RD access at the remote computer.

Always ask for credentials

Check this option to ask every user to enter the credentials every time they load the connection to get access to the VPN.



When you check this option, the 'Authentication' tab will automatically disappear in the connection settings dialog.

4.1.4.3 Authentication

In the 'Authentication' tab you will must select the authentication method you prefer:

CONNECTIONS

General
Host
Authentication
Display
Programs
Local Resources
Experience

User name:

Password:

Domain:

? Main Back Apply Finish

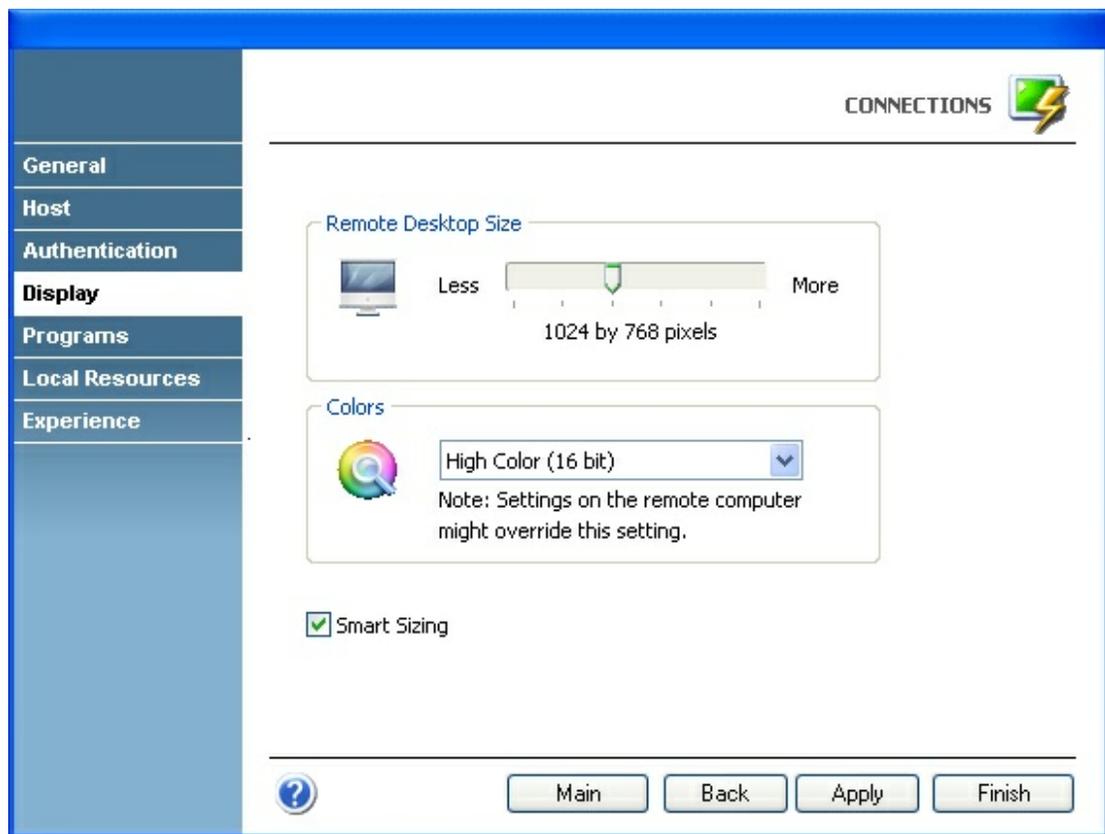
User Name
Enter the user name.

Password
Enter the password.

Domain
Enter the domain.

4.1.4.4 Display

In the 'Display' tab you will find the following parameters:



Remote Desktop Size

In this field you must enter a name the FTP connection.

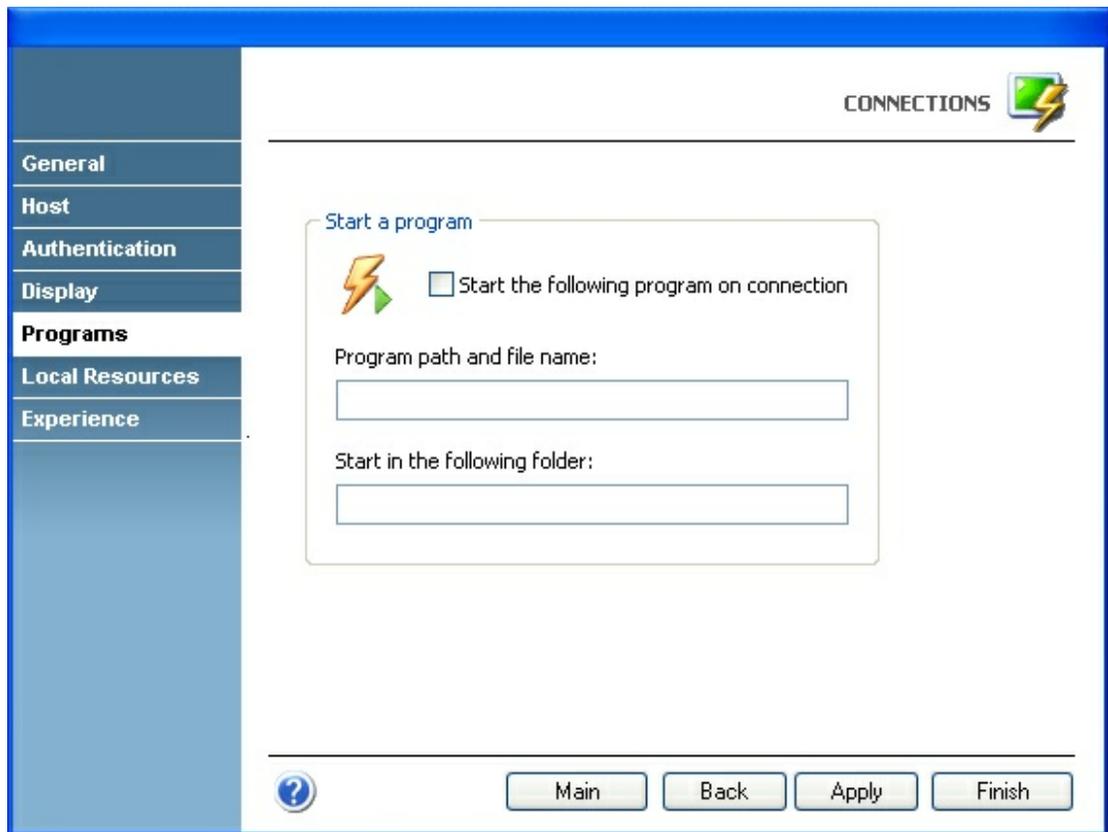
Colors

In this field you can type a description for the FTP connection.

Smart Sizing

4.1.4.5 Programs

In the 'Programs' tab you will find the following parameters to start a program when loading the connection:



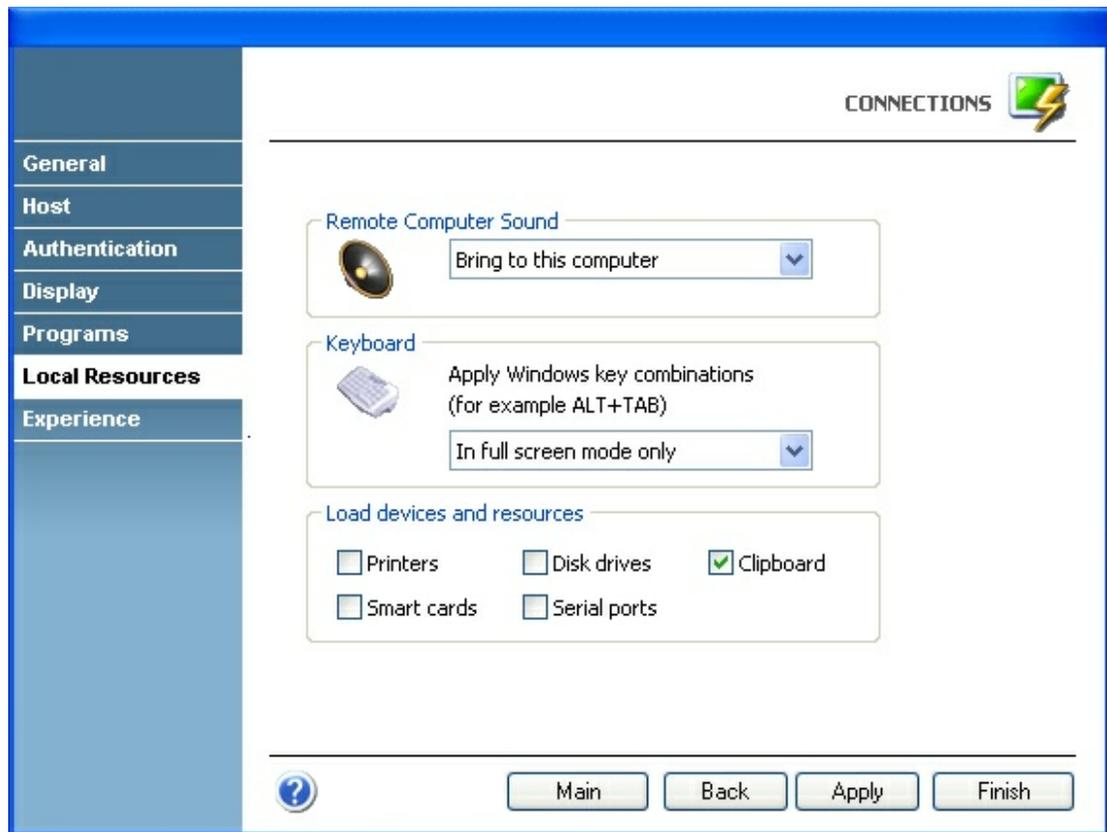
Start the following program on connection
Check this option to load the program.

Program Path and file name
Specify not only the program path but also the file name, for instance:
C:\Program Files\ProgramFolder\ProgramName.exe

Start in the following folder:
Specify here the folder where the program should be started.

4.1.4.6 Local Resources

In the 'Local Resources' tab you will find the following parameters:



Remote Computer Sound options

- Bring to this computer (selected by default)
- Do not play
- Leave at remote computer

Keyboard options

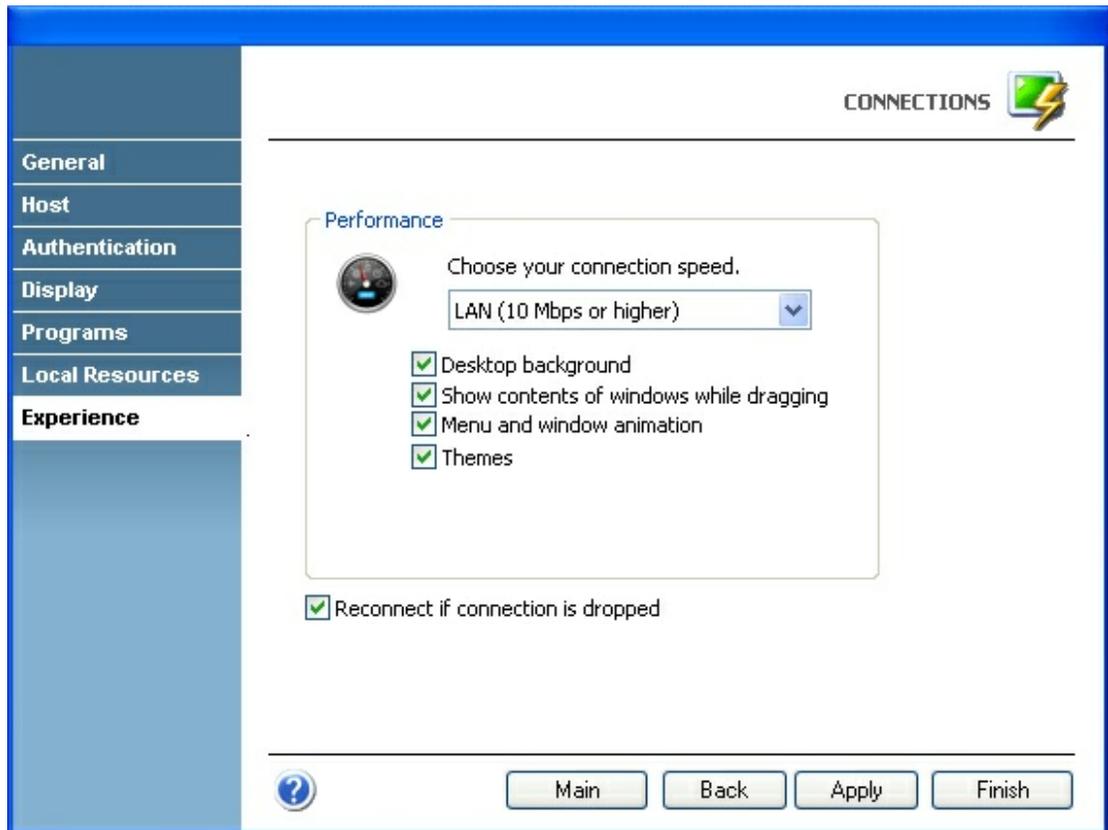
- Apply windows Key combinations in full screen mode only (selected by default)
- Apply windows Key combinations on the local computer
- Apply windows Key combinations on the remote computer

Load devices and resources (check to load)

- Printers
- Smart cards
- Disk drives
- Serial ports
- Clipboard (selected by default)

4.1.4.7 Experience

In the 'Experience' tab you will find the following parameters to determine the best performance:



Connection speed

- Modem (28.8 Kbps)
- Modem (56 Kbps)
- Broadband (128 Kbps - 1.5 Mbps)
- LAN (10 Mbps or higher)
- Custom

Desktop background

Select to see the remote computer's desktop background.

Show contents of windows while dragging

Select to see the content of windows while dragging.

Menu and window animation

Select to enable the menu and window animation.

Themes

Select to see the remote computer's themes.

Reconnect if connection is dropped

Check this option to reconnect when the session is closed.

4.1.5 VNC

For a detailed description of the parameters available when editing VNC connections, consult the following sections:

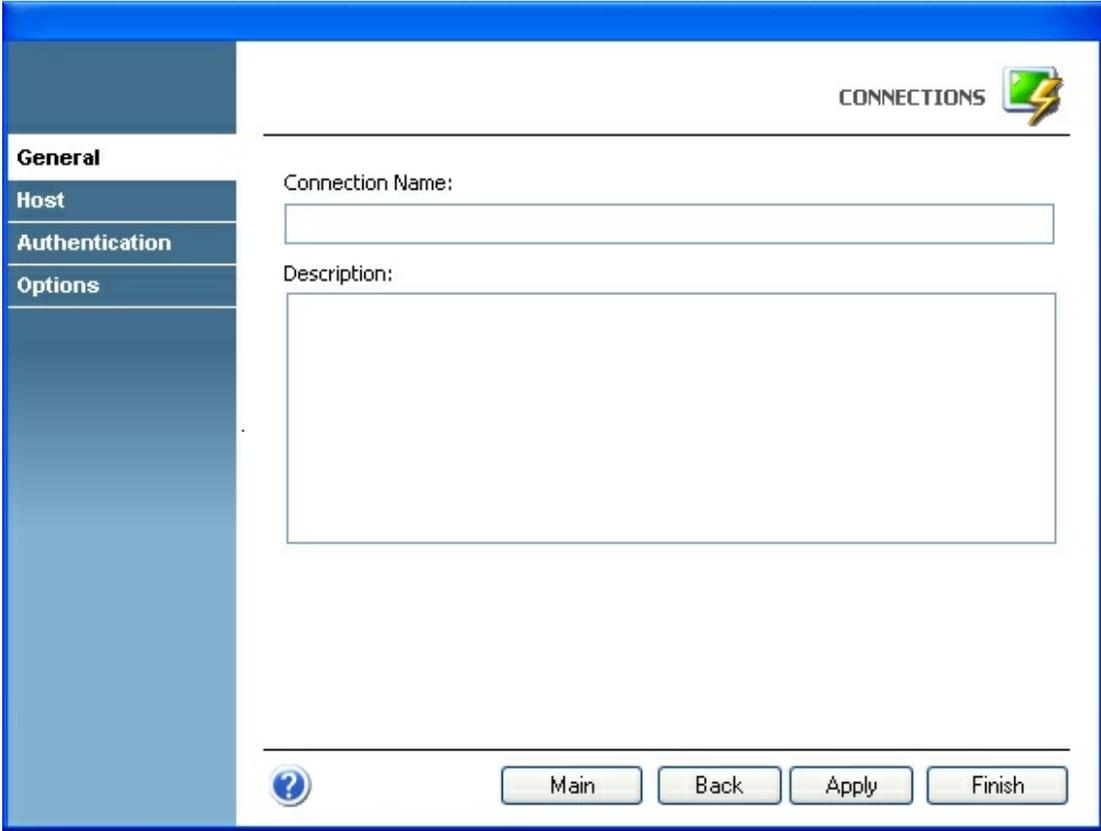
- [General](#)
- [Host](#)
- [Authentication](#)
- [Options](#)

Related Topics

Setting up a Connection

4.1.5.1 General

In the 'General' tab you will find the following parameters:



The screenshot shows a software window titled "CONNECTIONS" with a lightning bolt icon. On the left is a vertical navigation pane with four tabs: "General" (selected), "Host", "Authentication", and "Options". The main area contains two input fields: "Connection Name:" with a single-line text box, and "Description:" with a larger multi-line text box. At the bottom, there is a help icon (question mark in a circle) and four buttons: "Main", "Back", "Apply", and "Finish".

Connection name

In this field you must enter a name the FTP connection.

Description

In this field you can type a description for the FTP connection.

4.1.5.2 Host

In the 'Host' tab you will find the following parameters:

The screenshot shows a software window titled 'CONNECTIONS' with a lightning bolt icon. On the left is a vertical navigation pane with four tabs: 'General', 'Host', 'Authentication', and 'Options'. The 'Host' tab is currently selected. The main area of the dialog contains the following fields and controls:

- 'Address:' followed by a text input field.
- 'Port:' followed by a text input field.
- A checkbox labeled 'Always ask for credentials' which is currently unchecked.
- At the bottom left, there is a help icon (a question mark inside a circle).
- At the bottom right, there are four buttons: 'Main', 'Back', 'Apply', and 'Finish'.

Address

URL or IP address of the host machine that will act as the FTP server.

Port

TCP port defined for FTP access at the remote computer.

Always ask for credentials

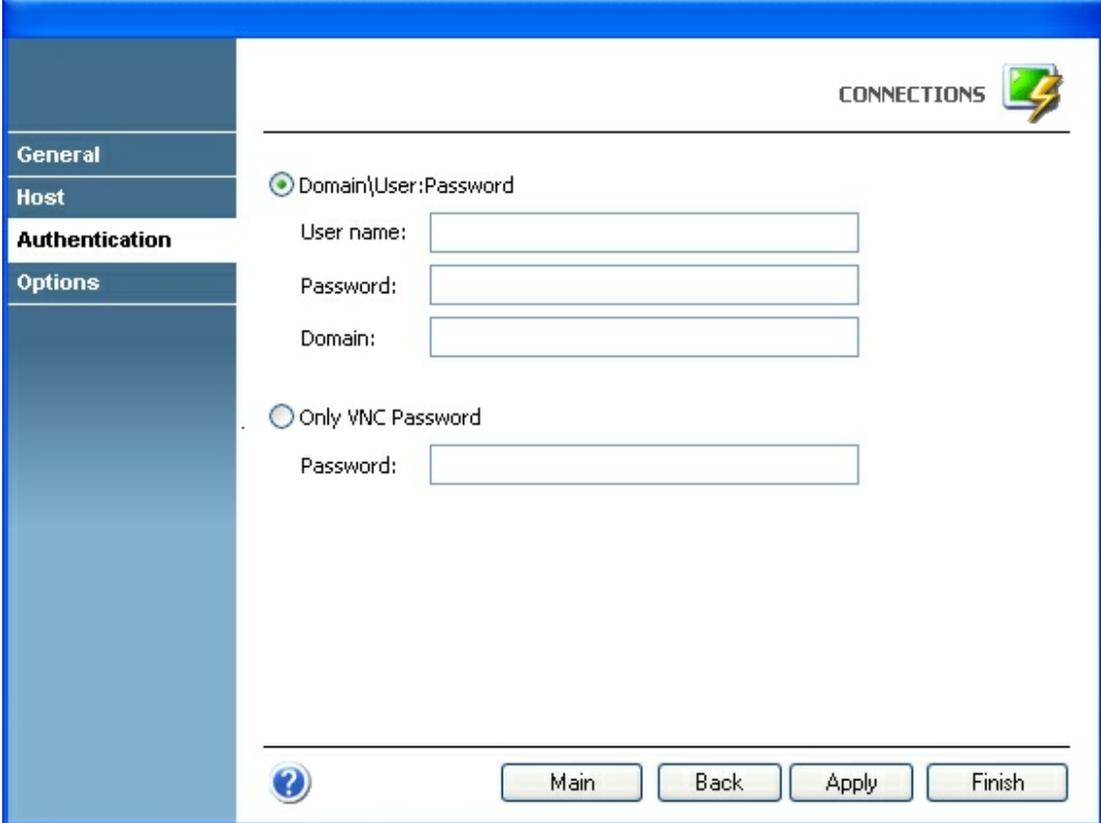
Check this option to ask every user to enter the credentials every time they load the connection to get access to the VPN.



When you check this option, the 'Authentication' tab will automatically disappear in the connection settings dialog.

4.1.5.3 Authentication

In the 'Authentication' tab you will must select the authentication method you prefer:



The screenshot shows a window titled "CONNECTIONS" with a lightning bolt icon. On the left is a navigation pane with tabs: "General", "Host", "Authentication" (selected), and "Options". The main area contains two radio button options for authentication. The first option, "Domain\User:Password", is selected and has three input fields for "User name:", "Password:", and "Domain:". The second option, "Only VNC Password", is unselected and has one input field for "Password:". At the bottom, there is a help icon (question mark) and four buttons: "Main", "Back", "Apply", and "Finish".

Domain\User:Pasword

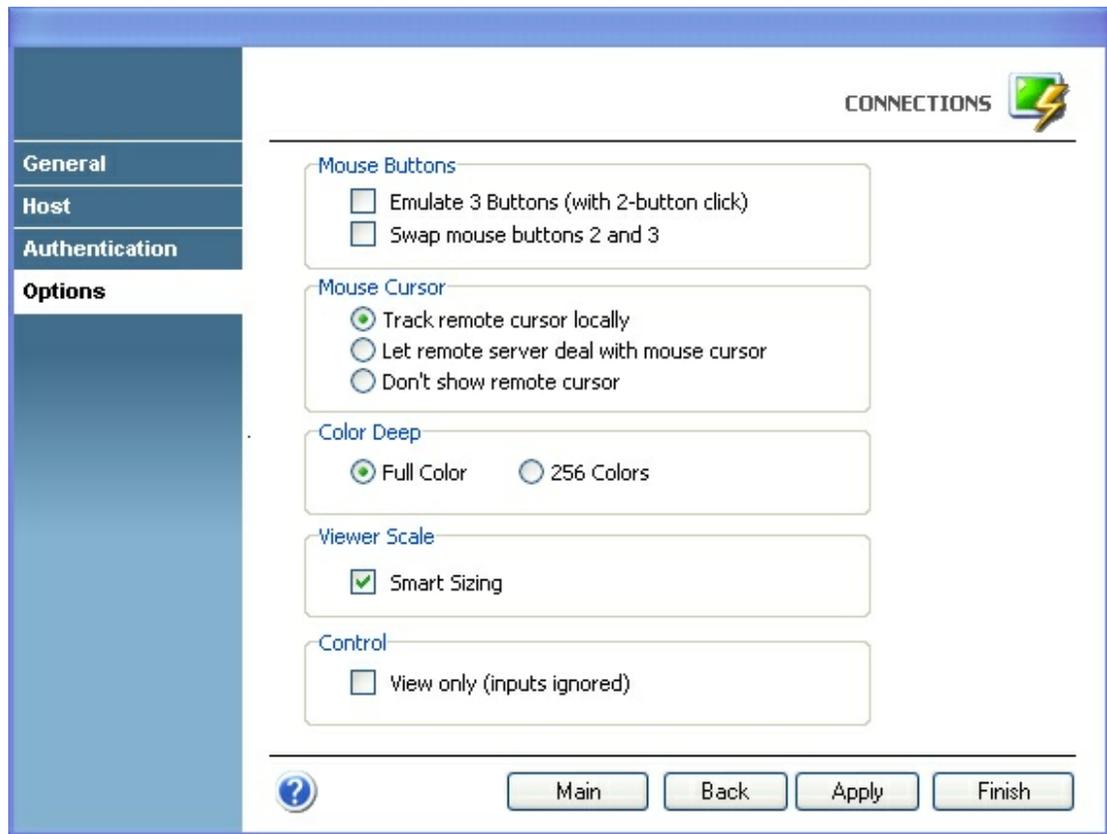
Enter the user name, the password and the domain data required to login.

Only VNC Password

Enter the VNC password required to login.

4.1.5.4 Options

In the 'Options' tab you will find the following parameters:



Mouse Buttons

Emulate 3 buttons (with 2-button click)
Swap mouse buttons 2 and 3

Mouse Cursor

Track remote cursor locally
Let remote server deal with mouse cursor
Don't show remote cursor

Color Deep

Full Color
256 Colors

Viewer Scale

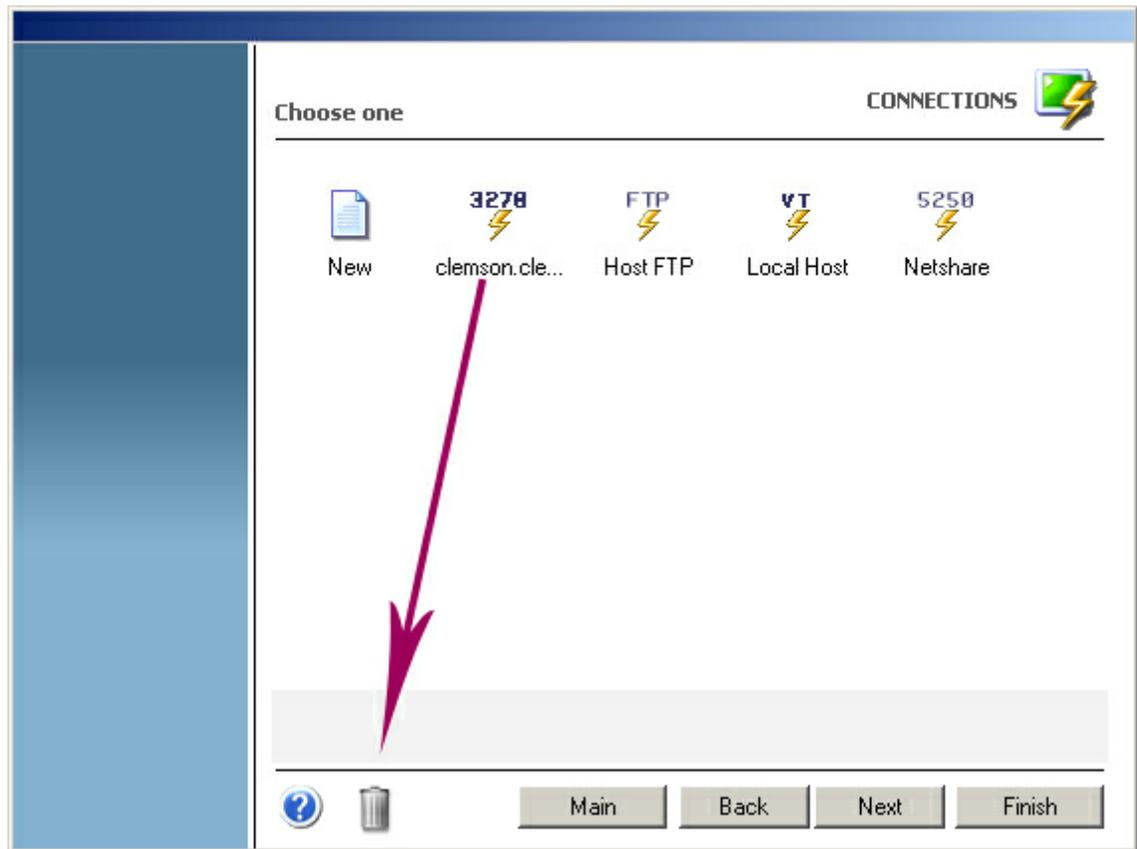
Smart Sizing

Control

View only (inputs ignored)

4.1.6 Deleting a Connection

To delete a connection, first select it and then click on the trash can icon. You can also drag and drop the connection you want to delete on the trash can icon.



 Connections deletion is not undoable, which means that once you delete a particular connection, it will be permanently lost.

4.2 Screen Styles

Screen Styles are managed from the 'Screen Styles' icon in the 'Settings' dialog.



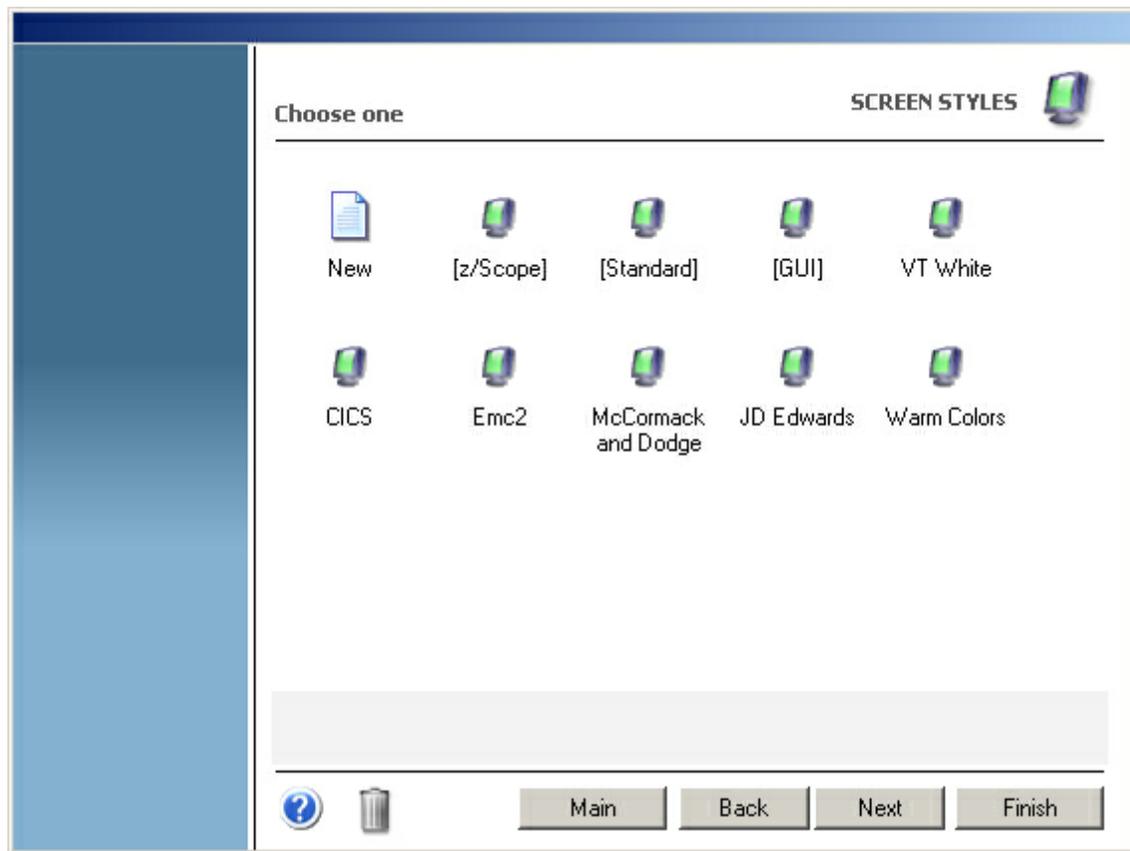
For more information about how to manage Screen Styles, consult the following sections:

- [Creating/Editing Screen Styles](#)
- [Deleting Screen Styles](#)

4.2.1 Creating/Editing Screen Styles

In the Screen Styles management dialog you will find a list of the existing Screen Styles. You will also find the 'New' icon which you can use to create a new Screen Style step by step.

If you want to modify the attributes of a previously created Screen Style, select it using the mouse and then click on 'Next', or just simply double-click on it.

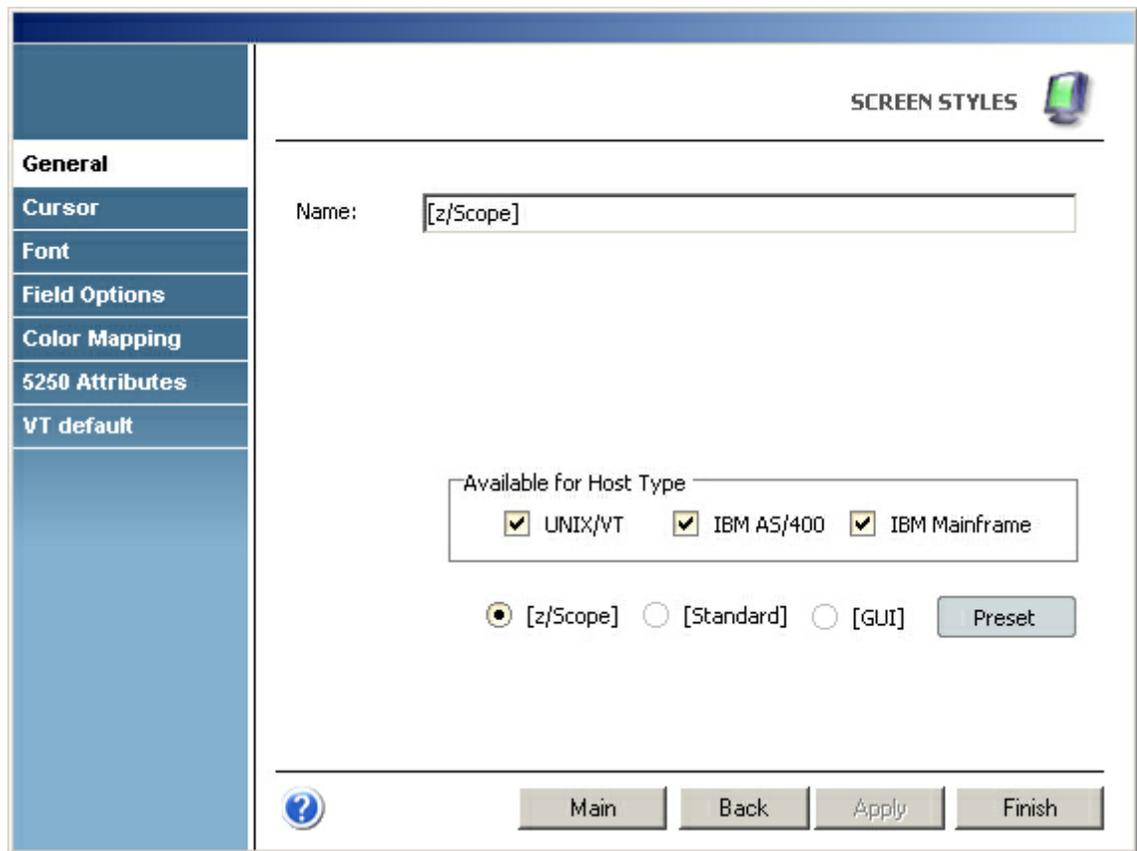


After you click on 'Next', you will be presented with a dialog in which you will have access to all the attributes of the Screen Style you are creating or editing. For a detailed description of these parameters, consult the following sections:

- [General](#)
- [Cursor](#)
- [Font](#)
- [Field Options](#)
- [Color Mapping](#)
- [5250 Attributes](#)
- [VT Defaults](#)

4.2.1.1 General

In the 'General' tab you will find the following parameters:



Name

In this field you must enter a name for the Screen Style.

Available for Host Type

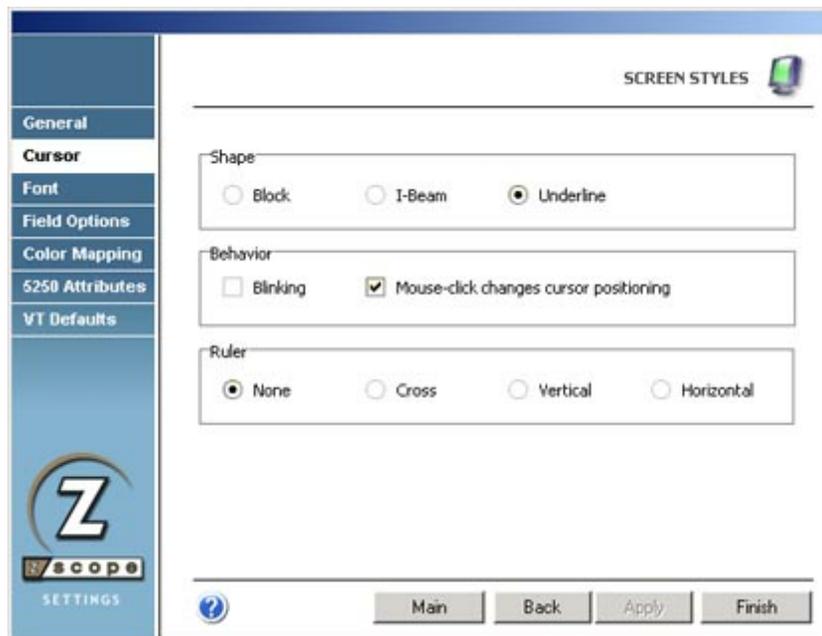
Make your Screen Style available for different type of hosts by marking the options below.

Preset

Select a base Screen Style to inherit its attributes to the new Screen Style.

4.2.1.2 Cursor

In the 'Cursor' tab you will find the following parameters:



Shape

Choose between three different cursor shapes: *Block*, *I-Beam* or *Underline*.

Behavior

Check the *Blinking* option to make the cursor blink.

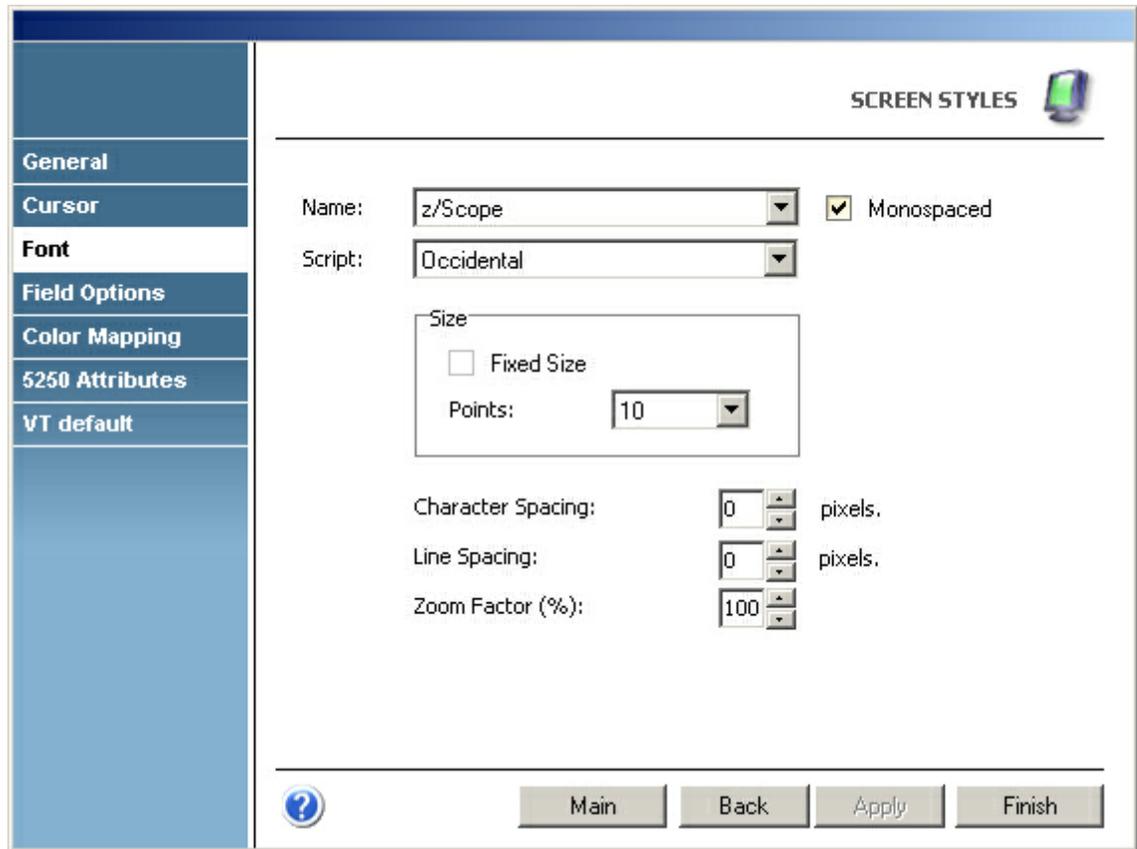
Check the *Mouse-click changes cursor positioning* option if you want to be able to change the current cursor position on the screen using the mouse.

Ruler

Specify if you want guidelines indicating the cursor's position on the screen. You can choose to display vertical, horizontal, or both (cross) guidelines. If you don't want to show guidelines, select the *None* option.

4.2.1.3 Font

In the 'Font' tab you will find the following parameters:



Name

Allows you to choose a font from the list of installed fonts. If the *Monospaced* option is checked, only monospaced fonts will be listed.

 Monospaced fonts are those in which all characters have the same width.

Script

Choose a code page for the selected font.

Size

Allows you to specify the size (in points) of the selected font. If the *Fixed size* option is checked, the screen font won't be resized when modifying the size of the window.

Character Spacing

Allows you to vary the character spacing of the selected font.

Line Spacing

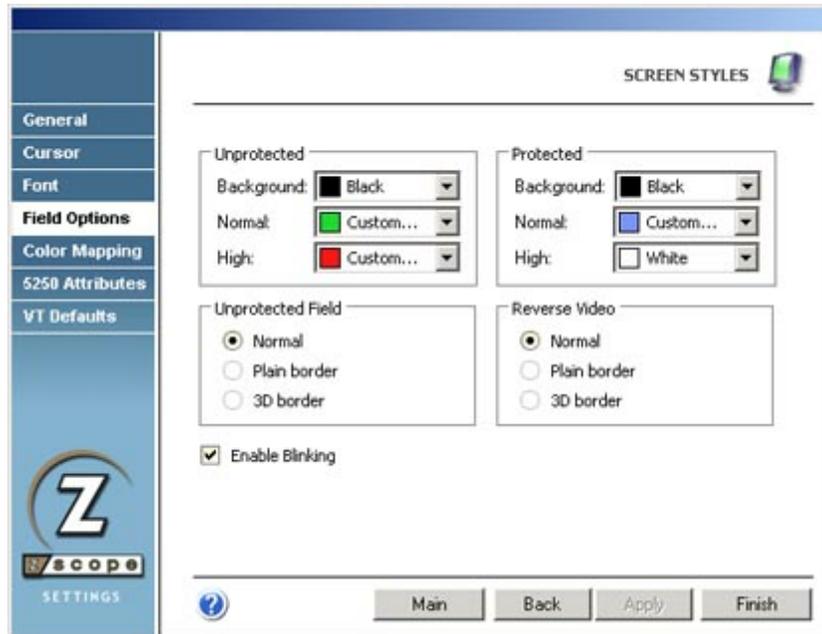
Allows you to vary line spacing for the screen style.

Zoom Factor (%)

Allows you to choose a level of zoom that will be applied in the emulation to each character of the font.

4.2.1.4 Field Options

In the 'Field Options' tab you will find the following parameters:



Unprotected

Allows to specify the background color and the foreground color for normal and high intensity unprotected fields.

Protected

Allows to specify the background color and the foreground color for normal and high intensity protected fields.

Unprotected Field

Sets normal, plain border or 3D style for unprotected (input) fields.

Reverse Video

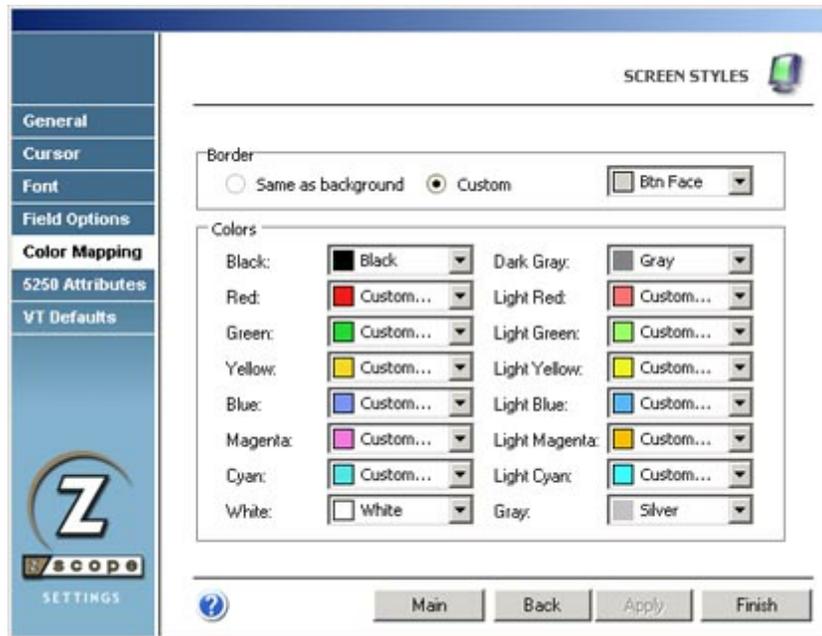
Sets normal, plain border or 3D style for reverse video attribute in the display emulation.

Enable Blinking

Check this option if you want the cursor to blink when positioned in an input field.

4.2.1.5 Color Mapping

In the 'Color Mapping' tab you will find the following parameters:



Border

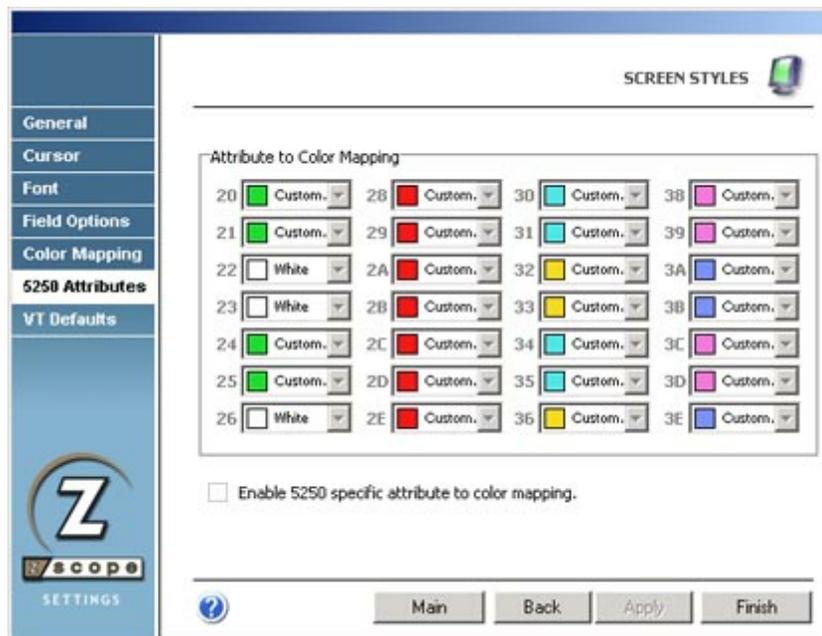
Specifies whether the border color will match the background ('Same as Background') or will have the color specified in the following combobox ('Custom').

Colors

Allows to remap the colors used for 3270 and 5250 extended attributes and VT/ANSI emulation.

4.2.1.6 5250 Attributes

In the '5250 Attributes' tab you will find the following parameters:

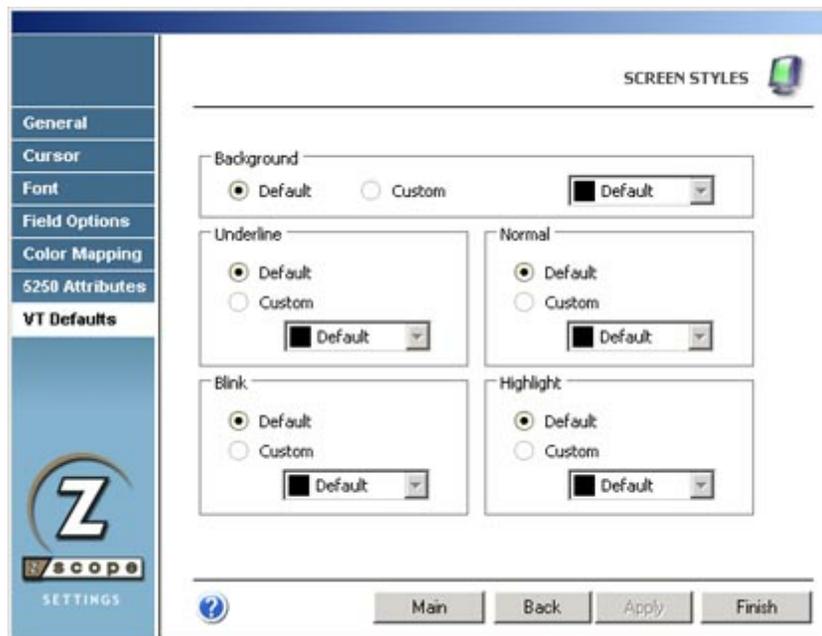


Attribute to Color Mapping

Allows you to map 5250 protocol specific attributes to a color. In order to be able to change these settings, the *Enable 5250 specific attributes to color mapping* option must be checked.

4.2.1.7 VT Defaults

In the 'VT Defaults' tab you will find the following parameters:



Background

Specifies the background color for VT terminal emulation. Default is black.

Underline

Specifies the foreground color assigned to underlined characters when no color attribute is specified by the VT host.

Normal

Specifies the foreground color assigned to low intensity characters when no color attribute is specified by the VT host.

Blink

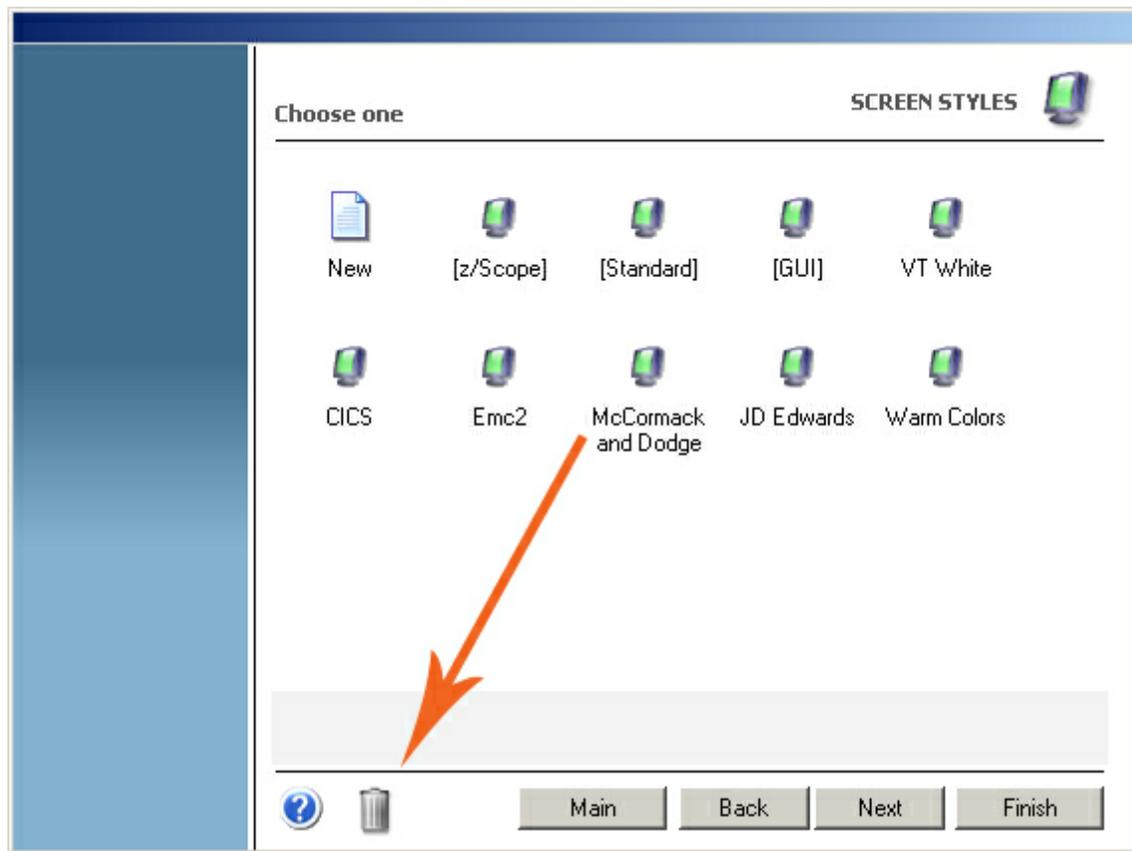
Specifies the foreground color assigned to blinking characters when no color attribute is specified by the VT host.

Highlight

Specifies the foreground color assigned to high intensity characters when no color attribute is specified by the VT host.

4.2.2 Deleting Screen Styles

To delete a Screen Style, first select it and then click on the trash can icon. You can also drag and drop the Screen Style you want to delete on the trash can icon.



 Screen Styles deletion is not undoable, which means that once you delete a particular Screen Style, it will be permanently lost.

4.3 Hotspots

Hotspots are managed from the 'Hotspots' icon in the 'Settings' dialog.



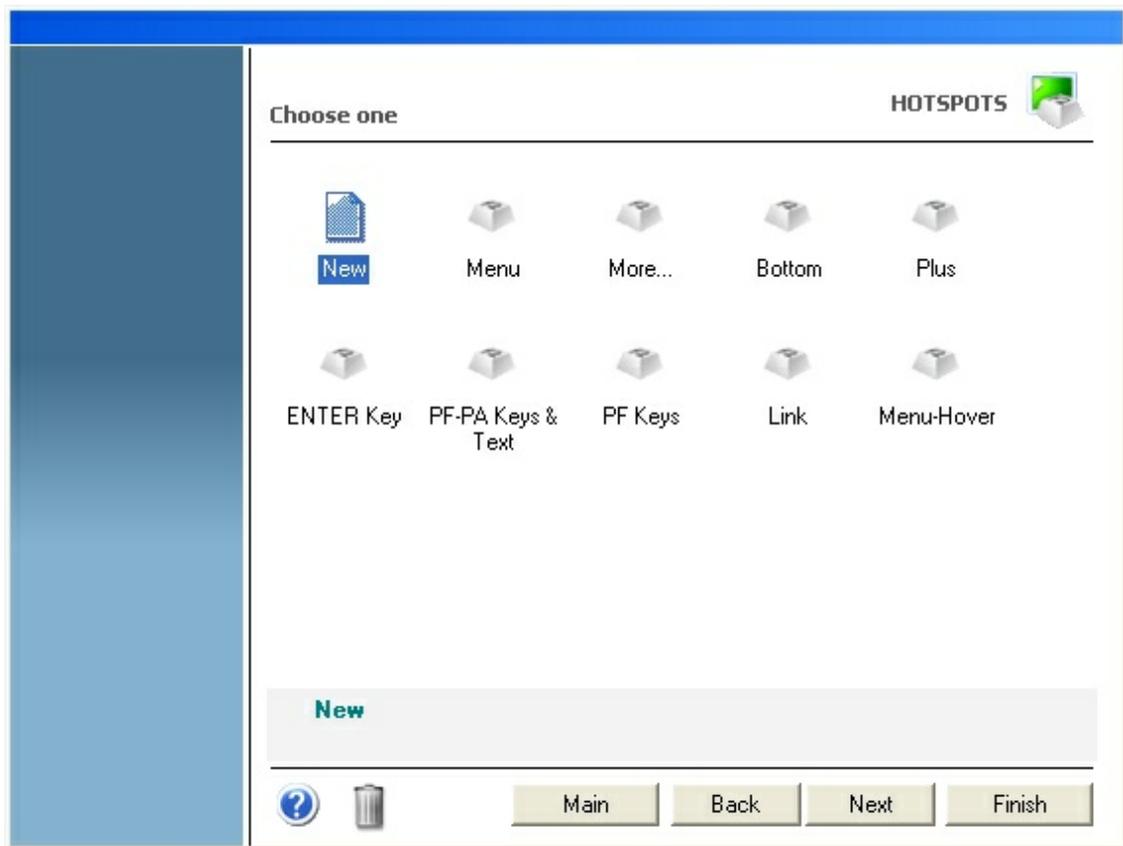
For more information about how to manage Hotspots, consult the following sections:

- [Creating/editing a Hotspot](#)
- [Deleting a Hotspot](#)

4.3.1 Creating/Editing a Hotspot

In the HotSpots management dialog you will find a list of the existing Hotspots. You will also find the 'New' icon, which you can use to create a new HotSpot.

If you want to modify the attributes of a previously created Hotspot, select it using the mouse and then click on 'Next', or just simply double-click on it.



After you click on 'Next', you will be presented with a dialog in which you will have access to all the attributes of the Hotspot you are creating or editing. For a detailed description of these parameters, consult the following sections:

- [General](#)
- [Rules](#)
- [Style](#)

 In order to be able to use your Hotspot within a Connection, you must first enable the Hotspot for that particular connection from the [Connections Settings](#) dialog.

4.3.1.1 General

In the 'General' tab you will find the following parameters:

The screenshot shows the 'HOTSPOTS' configuration window. On the left is a sidebar with three tabs: 'General', 'Rules', and 'Style'. The 'General' tab is selected. The main area contains a 'Name:' label followed by a text input field containing 'ENTER Key'. Below this is a section titled 'Available for Host Type' with three checkboxes, all of which are checked: 'UNIX/VT', 'IBM AS/400', and 'IBM Mainframe'. At the bottom of the window, there is a help icon (a question mark in a circle) and four buttons: 'Main', 'Back', 'Apply', and 'Finish'.

Name

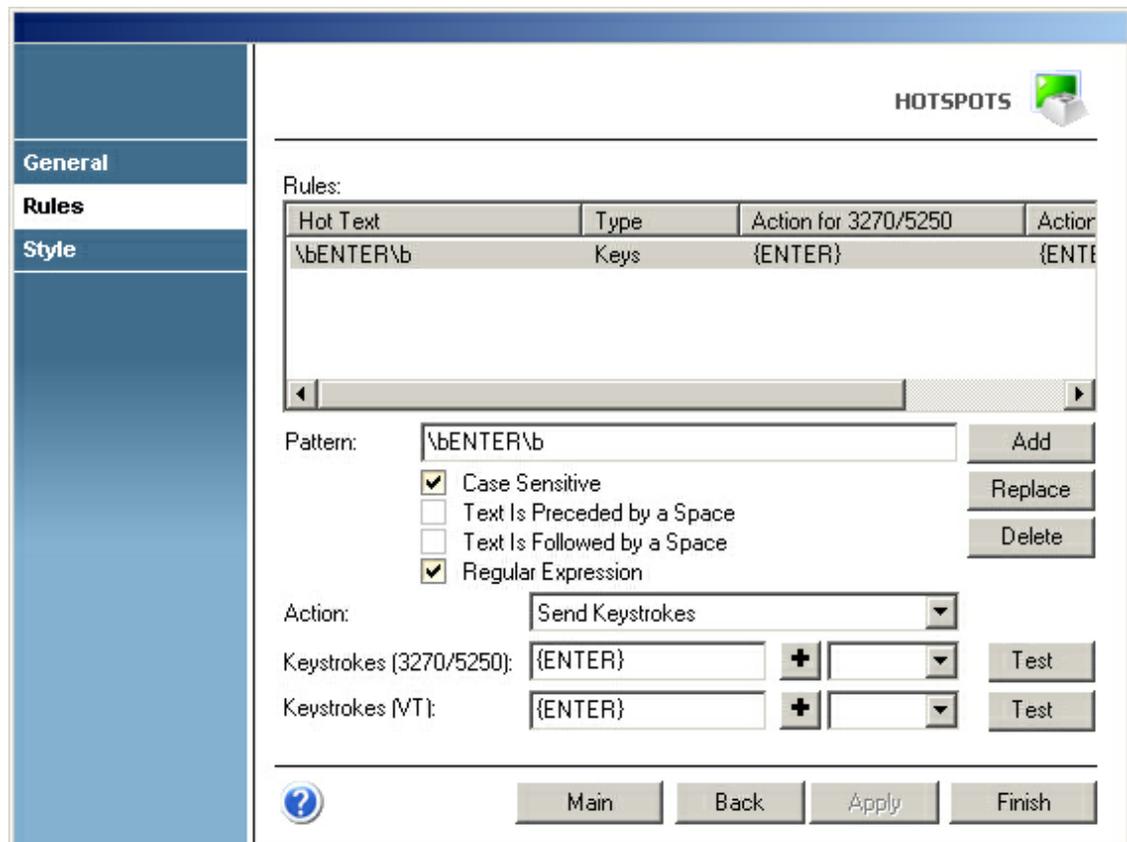
In this field you must enter a name for your Hotspot.

Available for Host Type

Make your Screen Style available for different type of hosts by marking the options below.

4.3.1.2 Rules

In the 'Rules' tab you will find the following parameters:



Rules grid

This grid contains the currently defined rules for the selected Hotspot.

Pattern

In this field you must type a rule in the form of a Regular Expression to match the screen string you want to turn into a HotSpot. If you are not familiar with Regular Expressions, see [Regular Expressions Appendix](#).

Case sensitive

Check this option if you want to consider the case when matching the Hot text.

Text is followed/preceded by a space

Check this options to specify that the Pattern will be matched only if preceded and/or followed by a blank character.

Action

This field specifies the action to be performed by the Hotspot when it is clicked by the user. Available options are:

- Send keystrokes: Select this option if you want to specify keystrokes to be sent in response to the HotSpot activation.
- Start a Macro: Select this option if you want to specify a Macro to be started in response to the HotSpot activation.

Keystrokes

In this field you can type the text that you want to be sent as keystrokes. You can also specify keys to be pressed; just select a key from the list and press the '+' button.



Note that keys must be enclosed in brackets (ie. {ENTER}).



This option is only available if the 'Send keystrokes' option was selected in the 'Action' field.

This is what the 'Rules' tab looks like when you select the 'start a macro' option:

The screenshot shows the 'Rules' tab in the zScope Terminal Emulator. The interface is divided into a left sidebar with 'General', 'Rules', and 'Style' tabs, and a main content area. The 'Rules' tab is active, displaying a table of rules. The table has four columns: 'Hot Text', 'Type', 'Action for 3270/5250', and 'Action'. A single rule is listed with the hot text '\bENTER\b', type 'Keys', and actions '{ENTER}' and '{ENTE'. Below the table, there is a 'Pattern' input field containing '\bENTER\b', and three checkboxes: 'Case Sensitive' (checked), 'Text Is Preceded by a Space' (unchecked), and 'Text Is Followed by a Space' (unchecked). There are also three buttons: 'Add', 'Replace', and 'Delete'. The 'Action' dropdown menu is open, showing 'Start a Macro' as the selected option. Below the dropdown is a 'Name' input field and an 'Open...' button. At the bottom of the window, there are four buttons: 'Main', 'Back', 'Apply', and 'Finish'.

Hot Text	Type	Action for 3270/5250	Action
\bENTER\b	Keys	{ENTER}	{ENTE

Name

In this field you can type the name of a Macro to be started in response to the Hotspot activation. You can use the 'Open' button to select a macro from the configuration directory.



This option is only available if the 'Start a macro' option was selected in the 'Action' field.

- To add a new Rule to the grid, follow these steps:

- In the 'Pattern' field, enter a Regular Expression to match the string you want to turn into a HotSpot, along with the adequate case and blank options.
- Select the action to be performed by the Hotspot when it is clicked by the user. Either choose the 'Send Keystrokes' or the 'Start a macro' option.

3. Type the keystrokes or the name of a macro according to the action selected before.



To add a key, select it from the key list and press the 'Add' button next to it. Keys can also be entered manually by enclosing them in brackets (ie. {ENTER}).

4. Click on the 'Add' button.

The new Rule you defined will now appear in the Rules grid.

• To modify an already defined Rule, do the following:

1. Select the Rule you want to modify from the Rules grid.
2. Modify the already defined 'Pattern' and 'Action' parameters as desired.
3. Click on the 'Replace' button to apply the changes to the Rule.

• To delete a Rule from the Rules grid, proceed this way:

1. Select the Rule that you want to delete from the Rules grid.
2. Click on the 'Delete' button.

The Rule you selected will be removed from the Rules grid.

• To test the results of a simulated screen text string, follow this steps:

1. Click on the 'Test' button. The 'Test Hotspot Rule' dialog will be displayed.



2. In the 'Text string' field, type the screen text simulation to be evaluated.
3. Click on the 'Test' Button. If the string you typed matches the current rule specified for the Hotspot, a message informing the action to be performed in response to the Hotspot activation will be displayed.
4. Click on the 'Exit' button to close this dialog and return to the 'Rules' tab.



This option is only available if the 'Send Keystrokes' option was selected in the 'Action' field.

Related Topics

[Appendix D - Regular Expressions](#)

4.3.1.3 Style

In the 'Style' tab you will find the following parameters:

The screenshot shows the 'Style' configuration window for a hotspot. The window has a blue header with the 'HOTSPOTS' logo. On the left is a sidebar with three tabs: 'General', 'Rules', and 'Style'. The 'Style' tab is selected. The main area contains three sections: 'Valid Area' with four spinners (First Row: 3, Last Row: 24, First Col.: 1, Last Col.: 132), 'Style' with a 'Show Format' dropdown set to 'Button', and 'Colors' with 'Foreground' and 'Background' dropdowns set to 'Button Text' and 'Button Face' respectively. Below these is an unchecked checkbox for 'Show in Toolbar'. At the bottom, there is a help icon, a 'Main' button, and three buttons: 'Back', 'Apply', and 'Finish'.

Valid Area

Specify the screen coordinates for the area that you want the Hotspot to work in.

Display Format

Choose among several display formats for the Hotspot from the drop-down list:

- None: The target string is left unaltered.
- Plain: The target string is replaced with a labeled plain button.
- Link: The target string is underlined web-link style.
- Button: The target string is replaced with a labeled 3D button.
- Hover: This works similarly to the 'Link' format, only that the target string is only underlined when the mouse pointer is dragged over it.

Colors

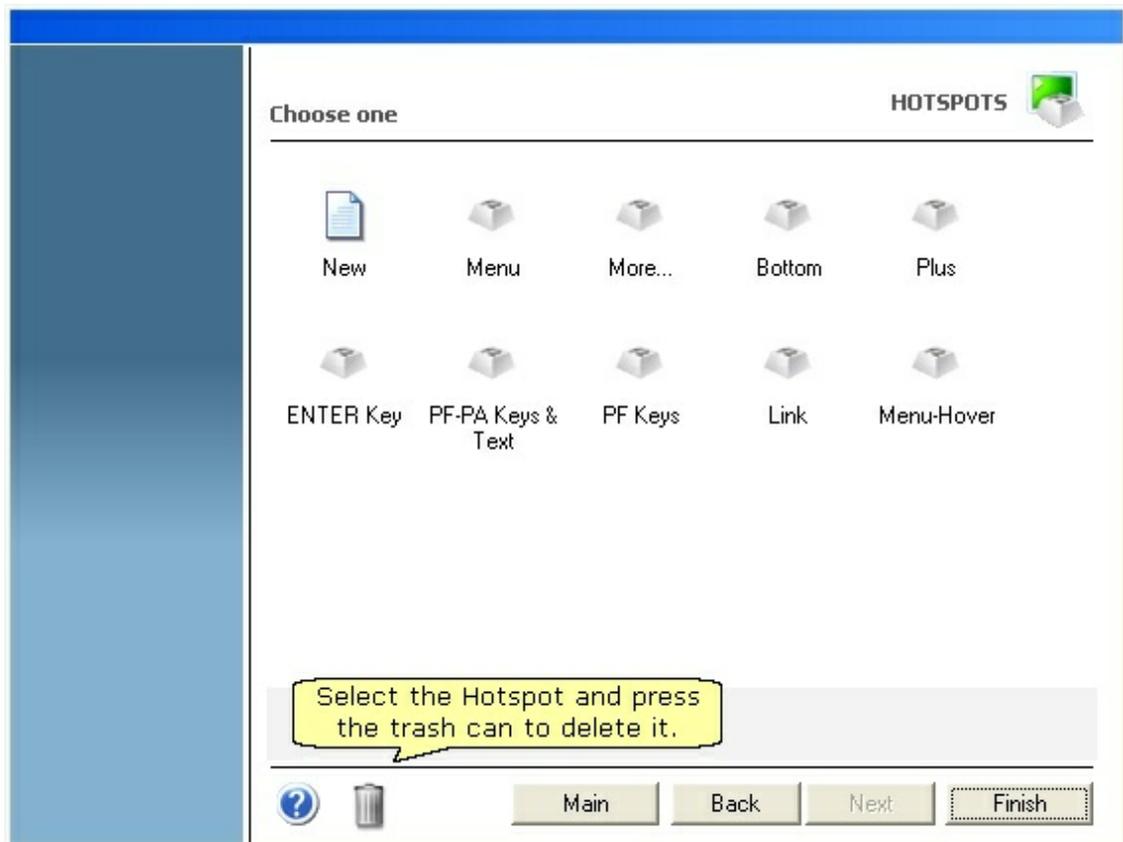
Choose a foreground and a background color for the Hotspot.

Show in Toolbar

Check this option if you want Hotspots to appear in a dynamic toolbar in the Emulation Display. See [Using Hotspots](#).

4.3.2 Deleting a Hotspot

To delete a Hotspot, first select it and then click on the trash can icon. You can also drag and drop the Hotspot you want to delete on the trash can icon.



 Hotspot deletion is not undoable, which means that once you delete a particular Hotspot, it will be permanently lost.

4.4 Keypads

Keypads are managed from the 'Keypad' icon in the 'Settings' dialog.



For more information about how to manage Keypads, consult the following sections:

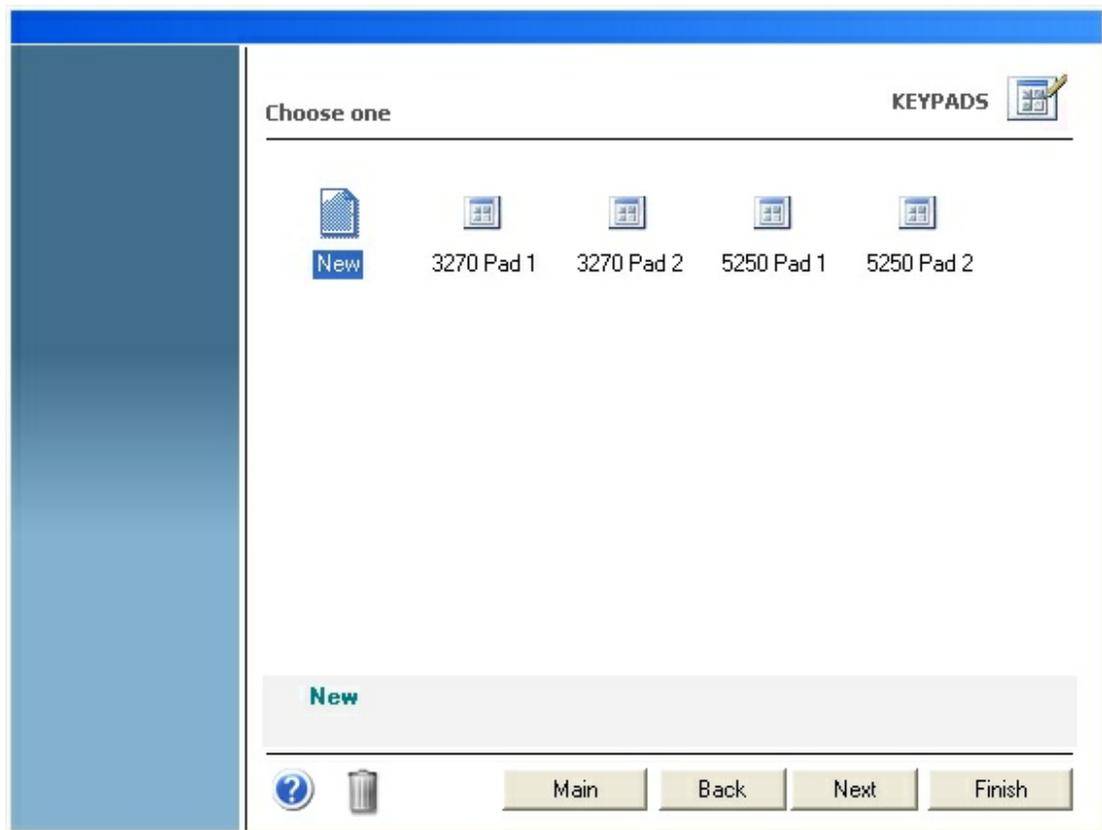
- [Creating/Editing a Keypad](#)
- [Deleting a Keypad](#)

 In order to be able to use your Keypad within a Connection, you must first enable the Keypad for that particular connection in the [Connections Settings](#) dialog.

4.4.1 Creating/Editing a Keypad

In the Keypads management dialog you will find a list of the existing Keypads. You will also find the 'New' icon which you can use to create a new Keypad from scratch.

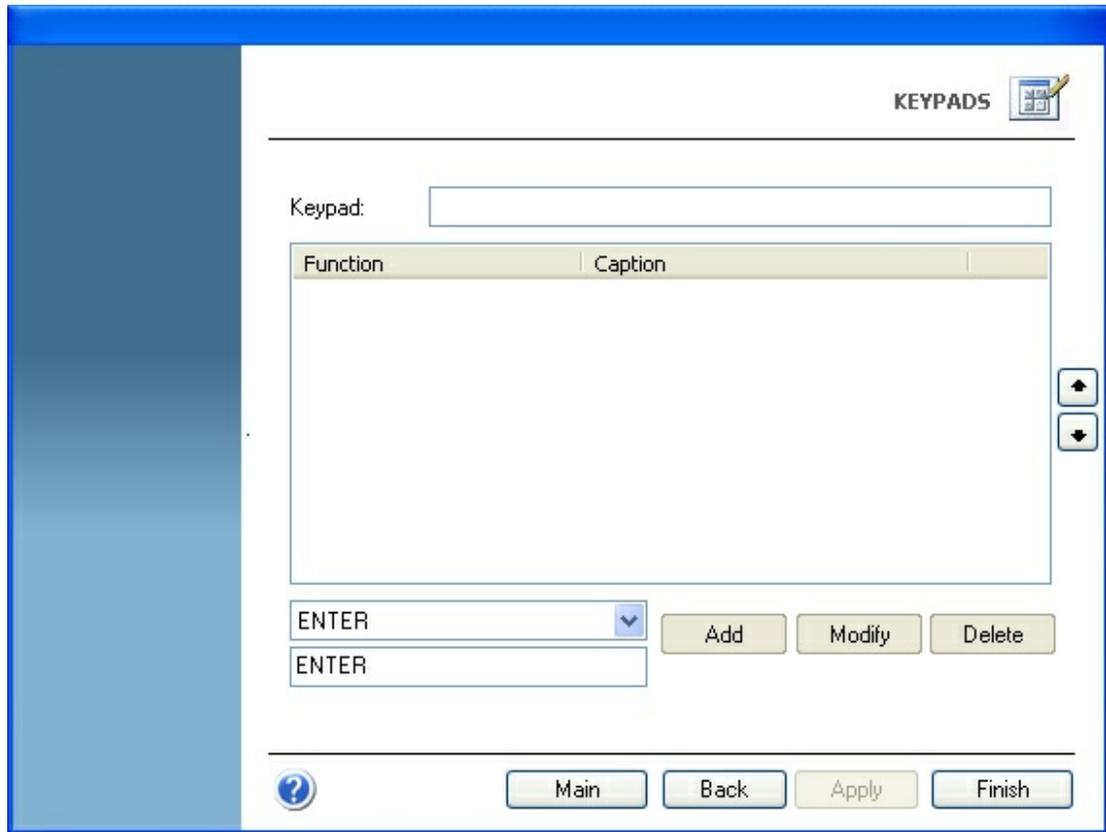
If you want to modify the attributes of a previously created Keypad, select it using the mouse and then click on 'Next', or just simply double-click on it.



 In order to be able to use your Keypad within a Connection, you must first enable the Keypad for that particular connection in the [Connections Settings](#) dialog.

Adding a keypad

Select the 'New' icon and after you click on 'Next', you will be presented with the 'Keypad Creation' dialog.



To add a new key to the Keypad, follow these steps:

1. Select a function to assign to the new key from the drop-down list.
2. Type a caption for the new key.
3. Click on the 'Add' button.

The new key you defined will now appear in the Keypad's key list. To modify the key's position within the Keypad, select it from the list and move it to the desired position using the key order selectors.

Modifying a keypad

To modify an already defined key, do the following:

1. Select the key you want to modify from the Keypad's key list.
2. Change the function and/or caption of the key.
3. Click on the 'Modify' button to apply the changes to the key.

Deleting a keypad

To delete a key from the Keypad, proceed this way:

1. Select the key you want to delete from the Keypad's key list.
2. Click on the 'Delete' button.

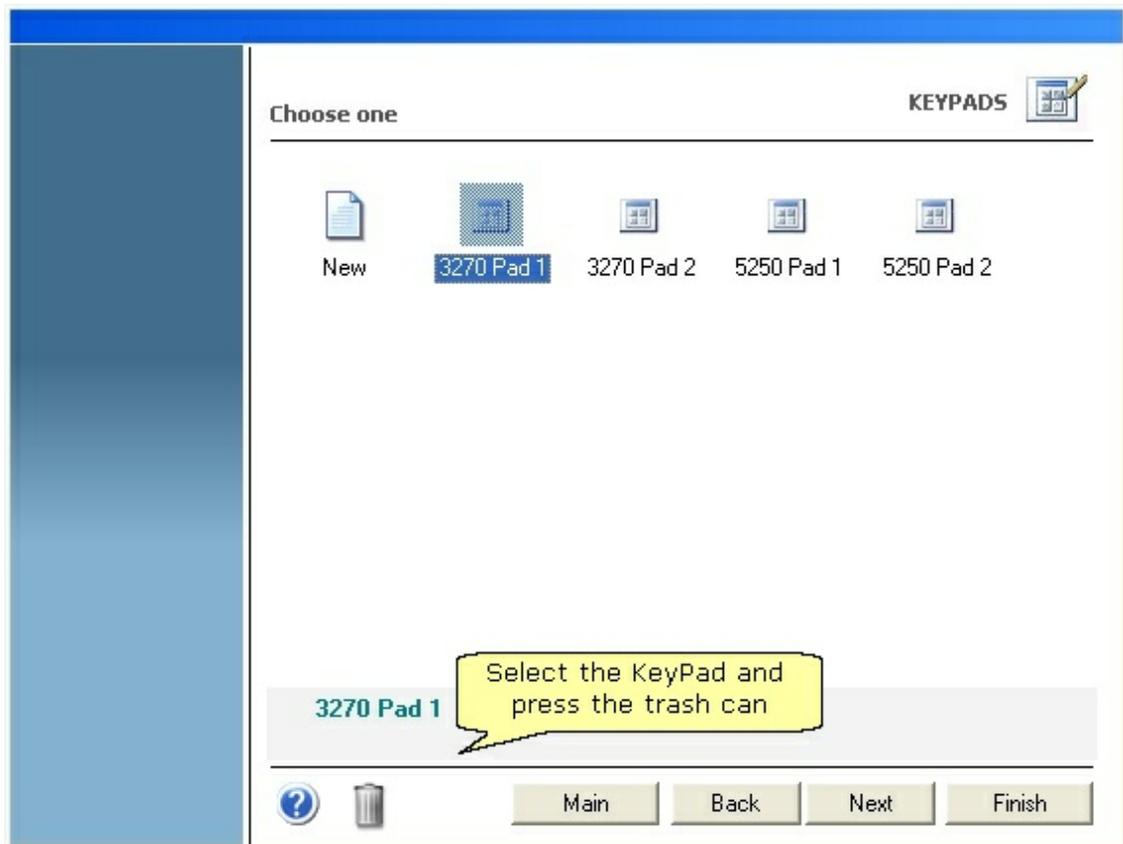
The key you selected will be removed from the Keypad's key list.

Read more

- [Deleting a Keypad](#)

4.4.2 Deleting a Keypad

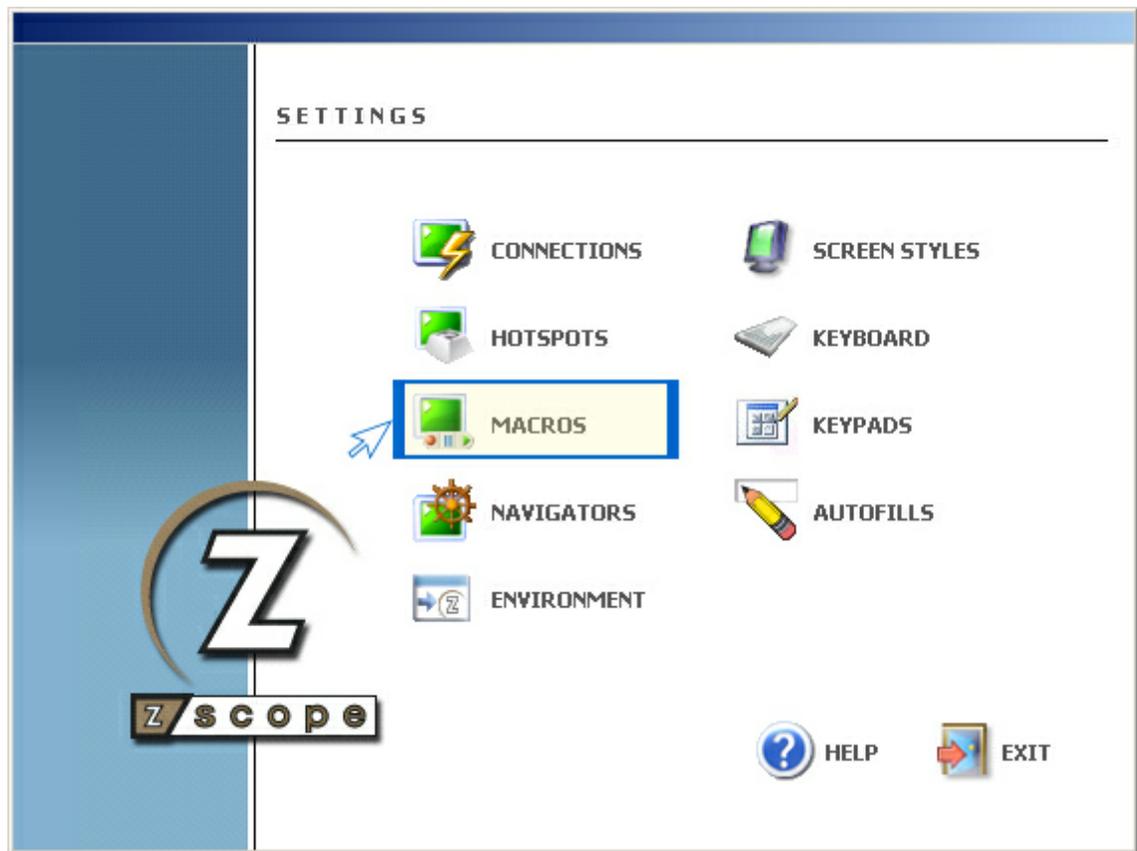
To delete a Keypad, first select it and then click on the trash can icon. You can also drag and drop the Keypad you want to delete on the trash can icon.



 Keypad deletion is not undoable, which means that once you delete a particular Keypad, it will be permanently lost.

4.5 Macros

Macros are managed from the 'Macros' icon in the 'Settings' dialog.



For more information about how to manage Macros, consult the following sections:

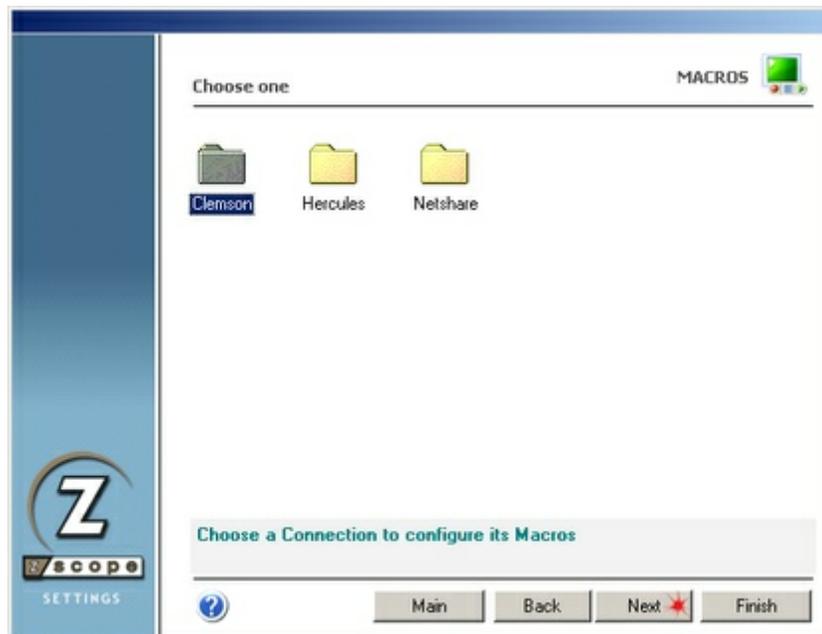
- [Editing Macros](#)
- [Debugging Macros](#)
- [Deleting Macros](#)

If you want to learn the simple steps to creating a macro from scratch and run it, please read this topic:

- [Using Macros](#)

4.5.1 Editing Macros

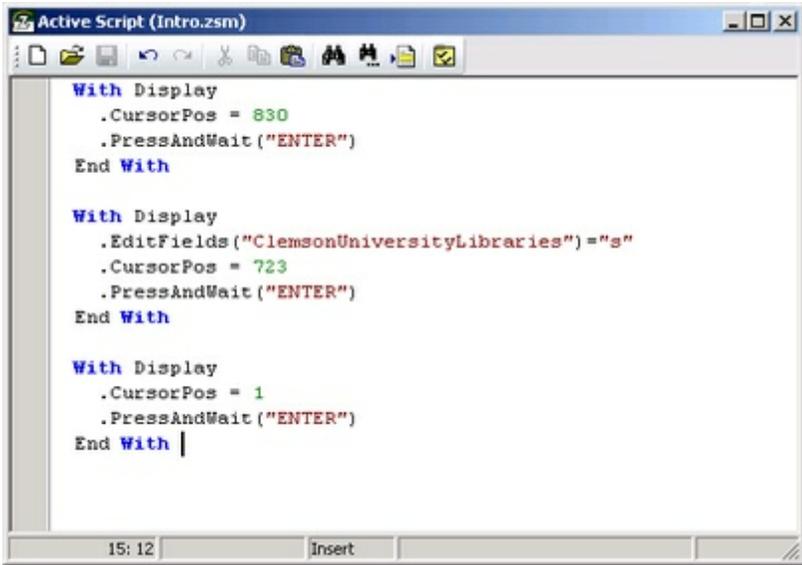
In the 'Macros Management' dialog, you will find a list of the existing connections.



Select a connection and then click on 'Next' to see all Macros defined for that particular connection.



If you want to modify a previously created Macro, select it and then click on 'Next'. You will see the 'Active Script' window which you can use to modify the Macro's code.



```

Active Script (Intro.zsm)
With Display
.CursorPos = 830
.PressAndWait ("ENTER")
End With

With Display
.EditFields ("ClemsonUniversityLibraries") ="s"
.CursorPos = 723
.PressAndWait ("ENTER")
End With

With Display
.CursorPos = 1
.PressAndWait ("ENTER")
End With |
  
```

- 
New Macro
Use this button to code a new macro from scratch.
- 
Open Macro
Use this button to open an existing macro file (.zsm).
- 
Save Macro
Use this button to save the changes you have made to the macro code. If you are creating a new macro, you will be prompted for a name for the macro file to be saved.
- 
Undo Changes
Use this button to rollback the last changes you have made to the macro code.
- 
Redo Changes
Use this button to recover changes previously discarded with the 'Undo' button.
- 
Cut
Use this button to cut the selected text-area.
- 
Copy
Use this button to copy the selected text-area into the clipboard buffer.
- 
Paste
Use this button to paste text from the clipboard into the screen at the cursor position.

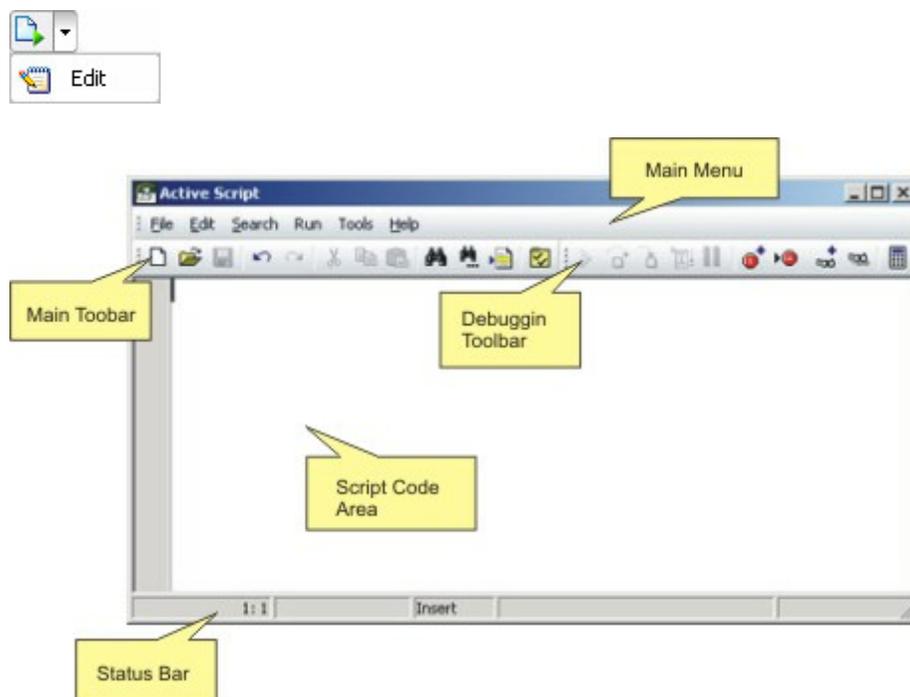
-  **Find Text**
Use this button to open the 'Find Text' window to search for a particular text string in your code.
-  **Find Next**
Use this button after the first matching string of your search has been found to find subsequent matching strings in your code.
-  **Go To Line Number**
Use this button to go to a specific line number in your code.
-  **Properties**
This button opens the 'Properties' window where you can customize several aspects of the Macro Editor.

Related Topics

[Using Macros](#)

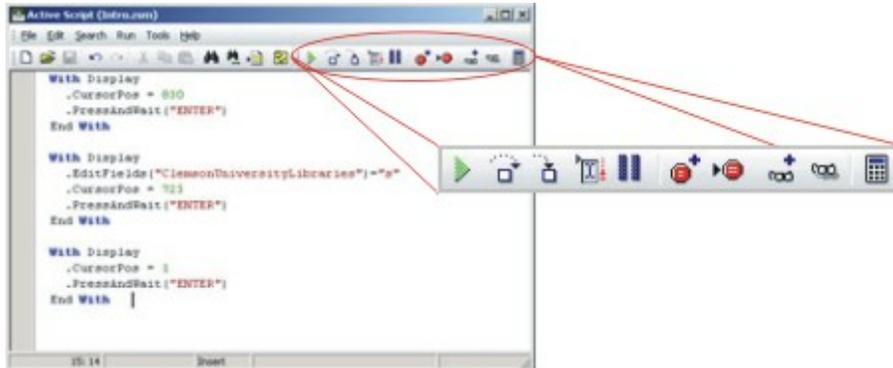
4.5.2 Debugging Macros

When a macro is created, it is saved as a script file that can be fully debugged using the 'Active Script' window in debugging mode. To access this mode, click on the 'Play Macro' button in the Main Toolbar and then select the 'Edit' option.



The 'Active Script' window will initially appear empty. Click on the 'Open' button in the toolbar  to display a dialog where you can choose the macro file (.zsm) want to debug.

When a macro file is opened, the macro debugging toolbar will become fully enabled. This will allow you to insert breakpoints, use step-into and step-over debugging techniques, add watches and also evaluate expressions.



-  **Start Or Resume Execution**
Use this button to start or resume code execution after it has been paused.
-  **Step Over**
Use this button to avoid executing a procedure line-by-line and execute it as a whole.
-  **Step Into**
Use this button to execute procedures line-by-line.
-  **Run To Cursor**
Use this button to execute all preceding code to the cursor position.
-  **Pause Execution**
Use this button to pause code execution.
-  **Add Breakpoint**
Use this button to set a breakpoint in the line you are currently positioned. Execution will pause every time it reaches the breakpoint.
-  **Breakpoints List**
Use this button to display a list of all established breakpoints in the macro and its properties.
-  **Add Watch**
Use this button to add a watch so you can monitor the value of a specific variable at any time in a real-time fashion.



Watches List

Use this button to display a list of all active watches with their current value.



Evaluate Expression

Use this button to query the value of any variable at any time.

Related Topics

[Using Macros](#)

4.5.3 Deleting Macros

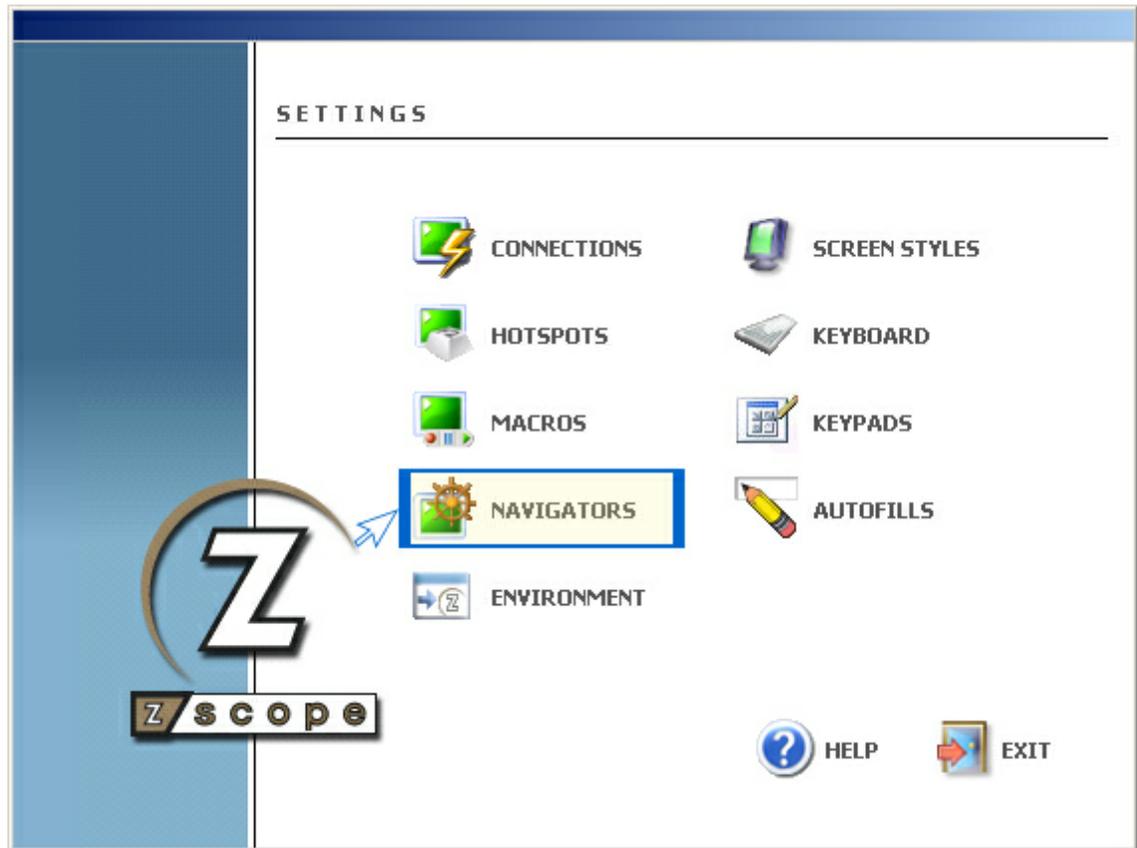
To delete a Macro, first select it and then click on the trash can icon. You can also drag and drop the Macro you want to delete on the trash can icon.



Macros deletion is not undoable, which means that once you delete a particular Macro, it will be permanently lost.

4.6 Navigators

Navigators are managed from the 'Navigators' icon in the 'Settings' dialog.

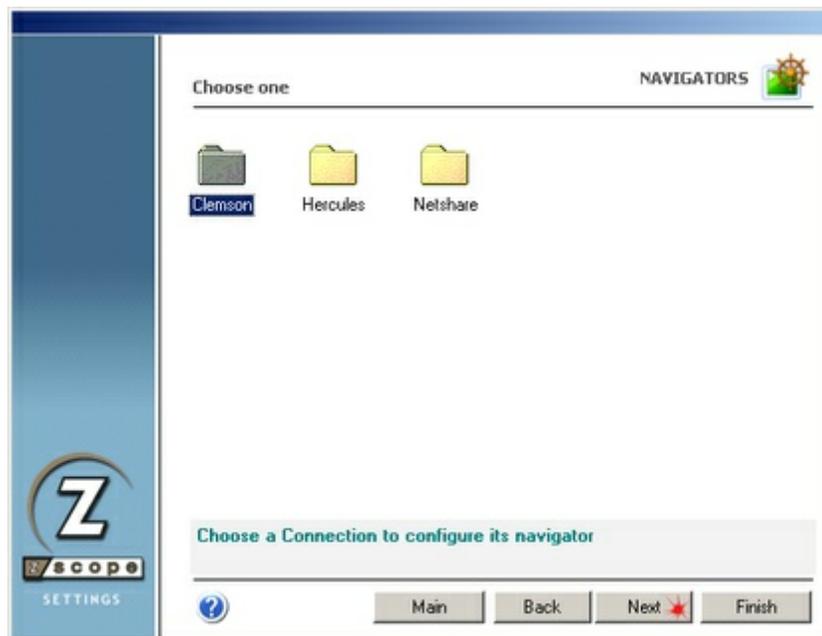


For more information about how to manage Navigators, consult the following sections:

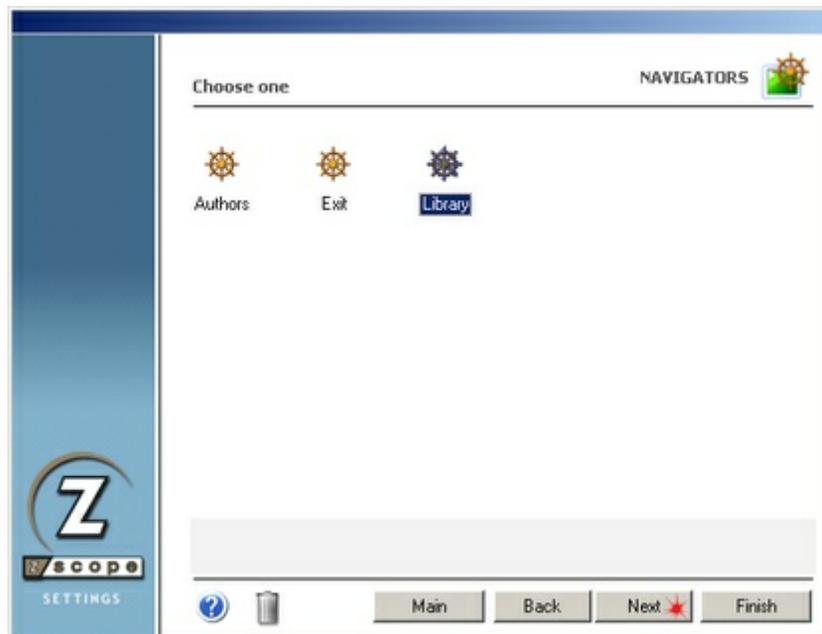
- [Creating a Navigator](#)
- [Editing a Navigator](#)
- [Deleting a Navigator](#)

4.6.1 Editing a Navigator

In the 'Navigators Management' dialog, you will find a list of the existing connections.



Select a connection and then click on 'Next' to see all Navigators defined for that particular connection.

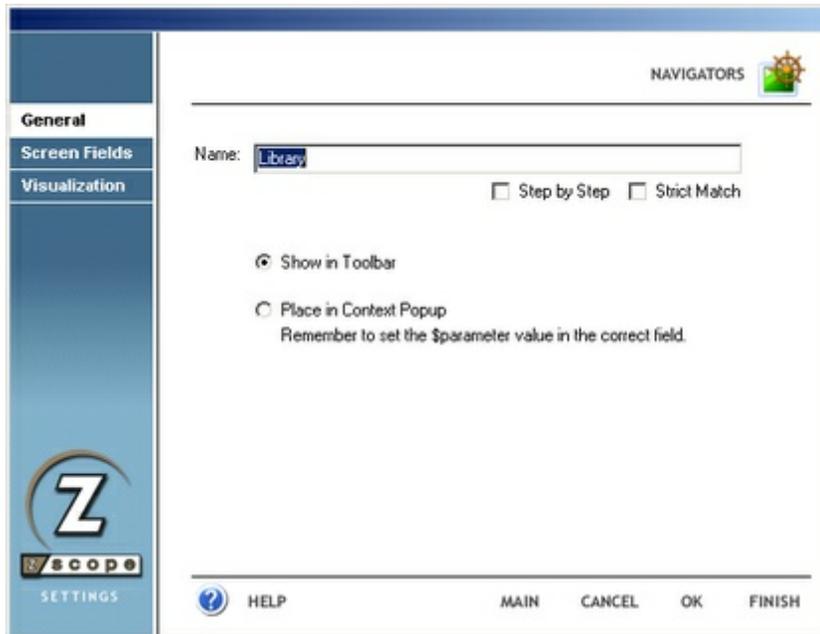


If you want to modify a previously created Navigator, select it and then click on 'Next'. You will be presented with a dialog in which you will have access to all the attributes of the Navigator you are editing. For a detailed description of these parameters, consult the following sections:

- [General](#)
- [Screen Fields](#)
- [Visualization](#)

4.6.1.1 General

In the 'General' tab you will find the following parameters:



Name

In this field you must enter a name for the Navigator.

Step By Step

When this option is checked, you will have to click on the Navigator button after each screen in the navigation sequence is shown.

Strict Match Only

If this option is checked, the internal recognition method employed to detect the screens that the Navigator is intended to work with will require a whole-page exact match. Otherwise, the match will be made only by the input field's caption, independently of the rest of the screen content.

Context Selection

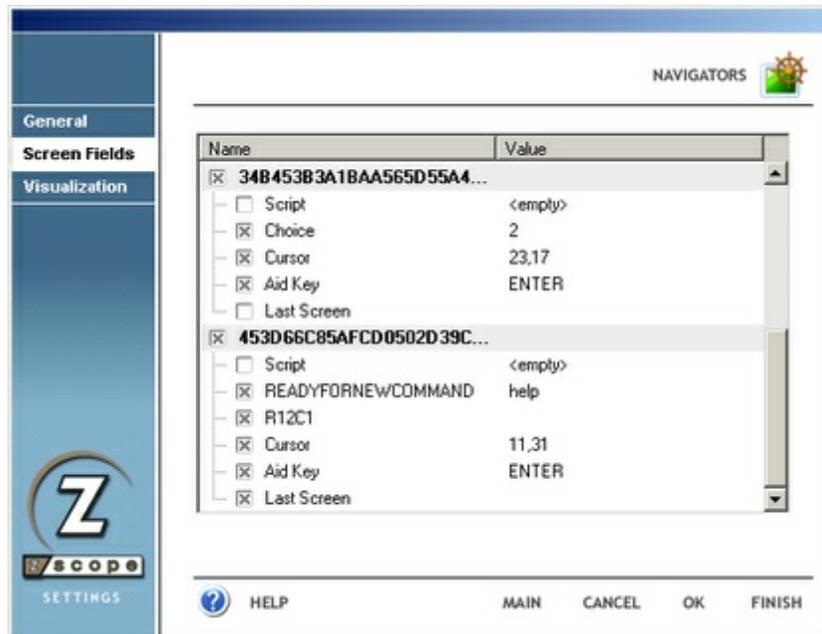
Select the 'Show in Toolbar' option if you want the Navigator to be visible inside a dynamic toolbar at the top of the Emulation Display.

Select the 'Place in Context Popup' option if you want the Navigator to be accessible from a context pop-up menu instead of from a button in a dynamic toolbar. See Running a Navigator (method #3).

 For this option to work, you must apply the 'Switch To Parameter' button to the desired fields in the 'Screen Fields' tab.

4.6.1.2 Screen Fields

In the 'Screen Fields' tab you will find the following parameters:



Screen unique ID number

This number is used to univocally identify the screen the Navigator applies to.

Clicking on this button while the screen ID number is selected will cause the Navigator to be discarded.

Script

Check this option if you want to add scripting code to the Navigator. See [Scripting Objects](#).

Click on this button to open the Active Script editor window where you will write your script. See [Editing Macros](#).

[Field_name]

This is the name of the input field that the Navigator applies to. Click on the 'Value' column to change the string that will be used to complete the field. You can also specify the following options:

Click on this button to specify that the Navigator will prompt the user for the string to be placed in the input

field instead of filling it automatically.

Switch To Parameter

Click on this button to specify that the text string selected from a context pop-up menu will be used to fill in the input field.



This button will only be enabled if the 'Place in Context Popup' option in the ['General' tab](#) is selected.

Switch To Encrypted Field

Click on this button if you want to encrypt the string in case it contains a password, a credit card number or any other kind of confidential information.

[Cursor]

These are the screen coordinates (row and column) of the input field to be filled by the Navigator.

Aid Key

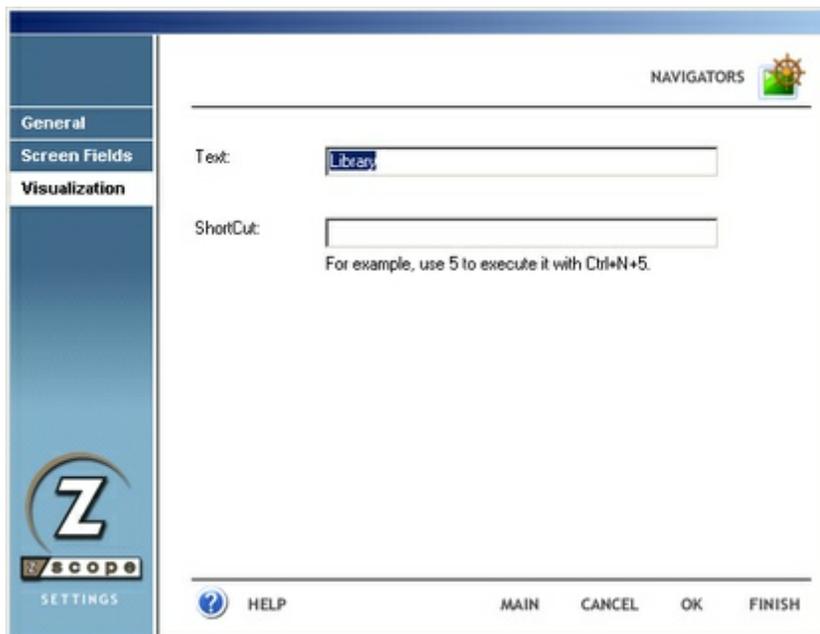
Specify a key to be sent after the input field is filled by the Navigator.

Last Screen

Indicates if the selected screen is the last of the navigation sequence.

4.6.1.3 Visualization

In the 'Visualization' tab you will find the following parameters:



Text

In this field you must enter a caption for the Navigator button.

Shortcut

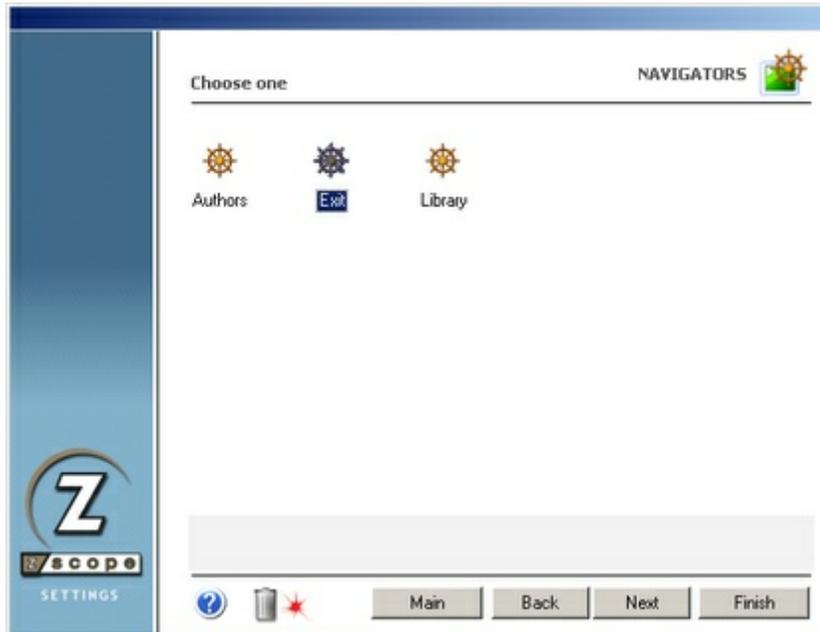
Trigger your Navigator with a custom combination of keys.

 Shortcuts are case sensitive.

 In order to work, the shortcut that you choose for a particular Navigator must be preceded by the Navigators feature global shortcut (Ctrl+N). If, for example, you assigned a particular Navigator the shortcut '5', in order to trigger this Navigator from its shortcut you must first press 'Ctrl+N' followed by '5'.

4.6.2 Deleting a Navigator

To delete a Navigator, first select it and then click on the trash can icon. You can also drag and drop the Navigator you want to delete on the trash can icon.



 Navigators deletion is not undoable, which means that once you delete a particular Navigator, it will be permanently lost.

4.7 Auto-Fills

Auto-Fills are managed from the 'Autofills' icon in the 'Settings' dialog.



For more information about how to manage Auto-Fills, consult the following sections:

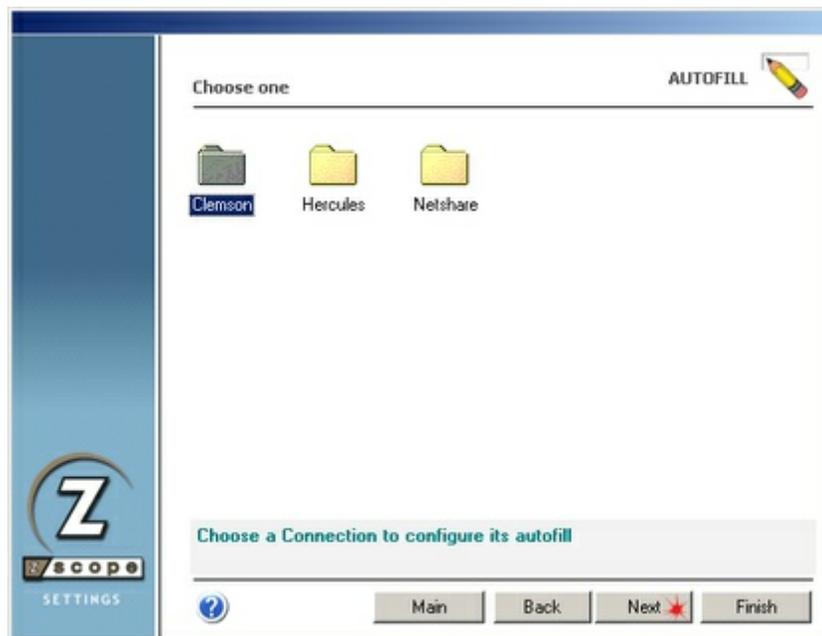
- [Editing an Auto-Fill](#)
- [Deleting an Auto-Fill](#)

Related Topics

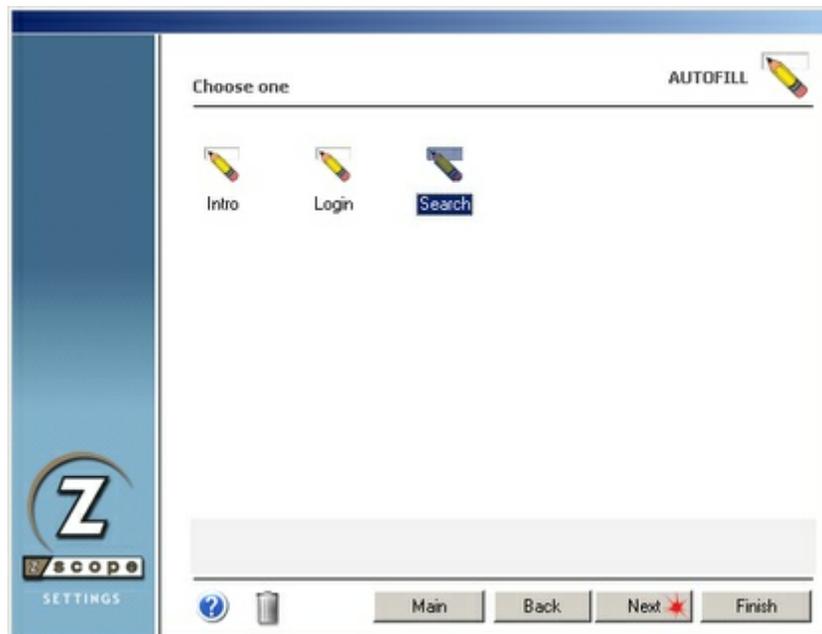
Overview of the Auto-Fills feature

4.7.1 Editing an Auto-Fill

In the Auto-Fills management dialog, you will find a list of the existing connections.



Select a connection and then click on 'Next' to see all Auto-Fills defined for that particular connection.

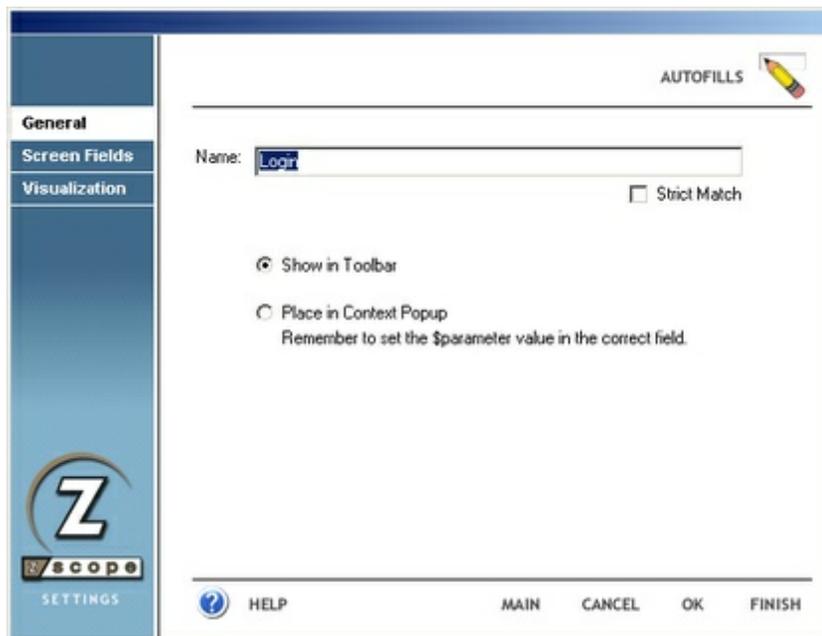


If you want to modify a previously created Auto-Fill, select it and then click on 'Next'. You will be presented with a dialog in which you will have access to all the attributes of the Auto-Fill you are editing. For a detailed description of these parameters, consult the following sections:

- [General](#)
- [Screen Fields](#)
- [Visualization](#)

4.7.1.1 General

In the 'General' tab you will find the following parameters:



Name

In this field you must enter a name for the Auto-Fill.

Strict Match Only

If this option is checked, the internal recognition method employed to detect the screen that the Auto-Fill is intended to work with will require a whole-page exact match. Otherwise, the match will be made only by the input field's caption, independently of the rest of the screen content.

Toolbar/Context Selection

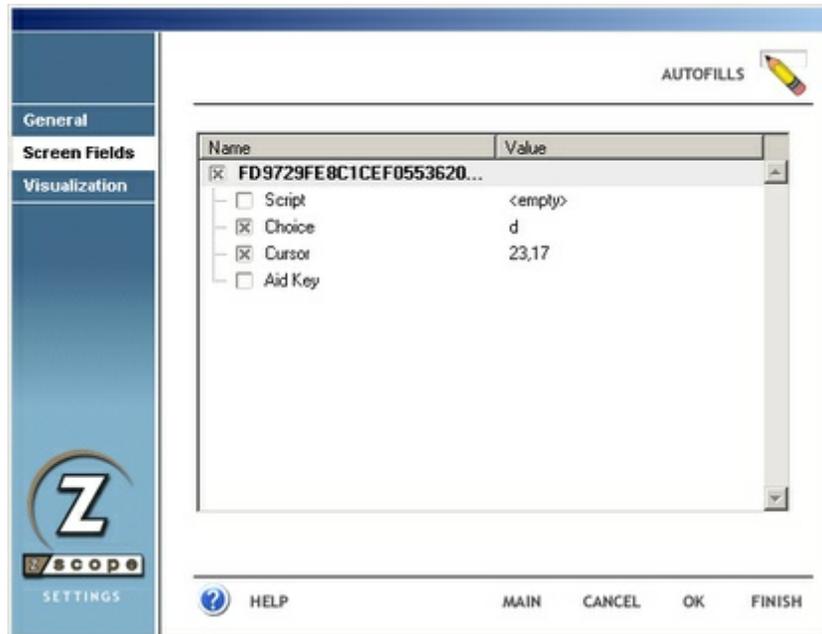
Select the 'Show in Toolbar' option if you want the Auto-Fill to be visible inside a dynamic toolbar at the top of the Emulation Display.

Select the 'Place in Context Popup' option if you want the Auto-Fill to be accessible from a context pop-up menu instead of from a button in a dynamic toolbar. See Using an Auto-Fill (method #3).

 For this option to work, you must apply the 'Switch To Parameter' button to the desired fields in the 'Screen Fields' tab.

4.7.1.2 Screen Fields

In the 'Screen Fields' tab you will find the following parameters:



Screen unique ID number

This number is used to univocally identify the screen the Auto-Fill applies to.

Delete Screen

Clicking on this button while the screen ID number is selected will cause the Auto-Fill to be discarded.

Script

Check this option if you want to add scripting code to the Auto-Fill. See [Scripting Objects](#).

Edit Script

Click on this button to open the Active Script editor window where you will write your script. See [Editing Macros](#).

[Field_name]

This is the name of the input field that the Auto-Fill applies to. Click on the 'Value' column to change the string that will be used to complete the field.

Switch To Ask For Input

Click on this button to specify that the Auto-Fill will prompt the user for the string to be placed in the input field instead of filling it automatically.

Switch To Parameter

Click on this button to specify that the text string selected from a context pop-up menu will be used to fill in the input field.



This button will only be enabled if the 'Place in Context Popup' option in the 'General' tab is selected.

Switch To Encrypted Field

Click on this button if you want to encrypt the string in case it contains a password, a credit card number or any other kind of confidential information.

[Cursor]

These are the screen coordinates (row and column) of the input field to be filled by the Auto-Fill.

Aid Key

Specify a key to be sent after the input field is filled by the Auto-Fill.

4.7.1.3 Visualization

In the 'Visualization' tab you will find the following parameters:



Text

In this field you must enter a caption for the Auto-Fill button.

Shortcut

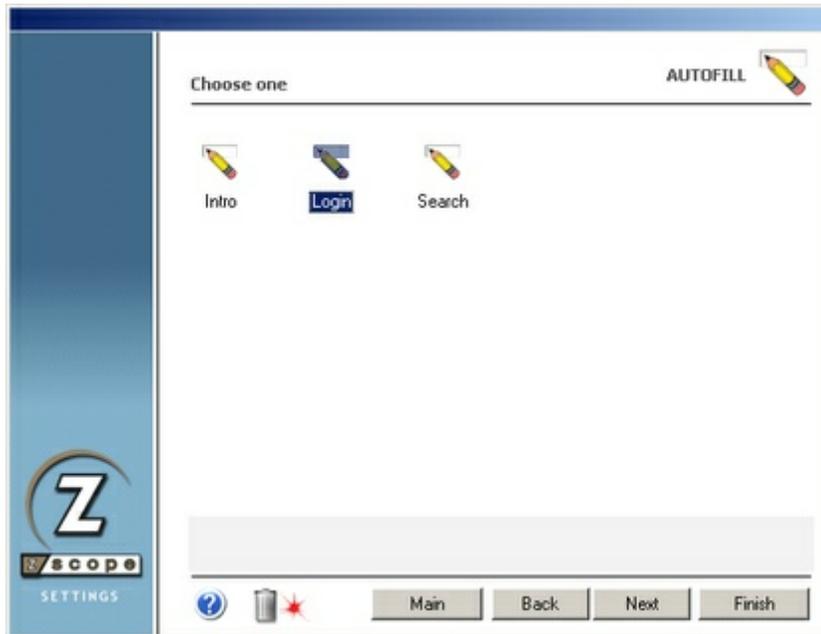
Specify a custom combination of keys to trigger the Auto-Fill.

 Shortcuts are case sensitive.

 In order to work, the shortcut that you choose for a particular Auto-Fill must be preceded by the Auto-Fill feature global shortcut (Ctrl+N). If, for example, you assigned a particular Auto-Fill the shortcut '5', in order to trigger this Auto-Fill from its shortcut you must first press 'Ctrl+N' followed by '5'.

4.7.2 Deleting an Auto-fill

To delete an Auto-Fill, first select it and then click on the trash can icon. You can also drag and drop the Auto-Fill you want to delete on the trash can icon.



 Auto-Fills deletion is not undoable, which means that once you delete a particular Auto-Fill, it will be permanently lost.

4.8 Keyboard

Keyboards are managed from the 'Keyboard' icon in the 'Settings' dialog.

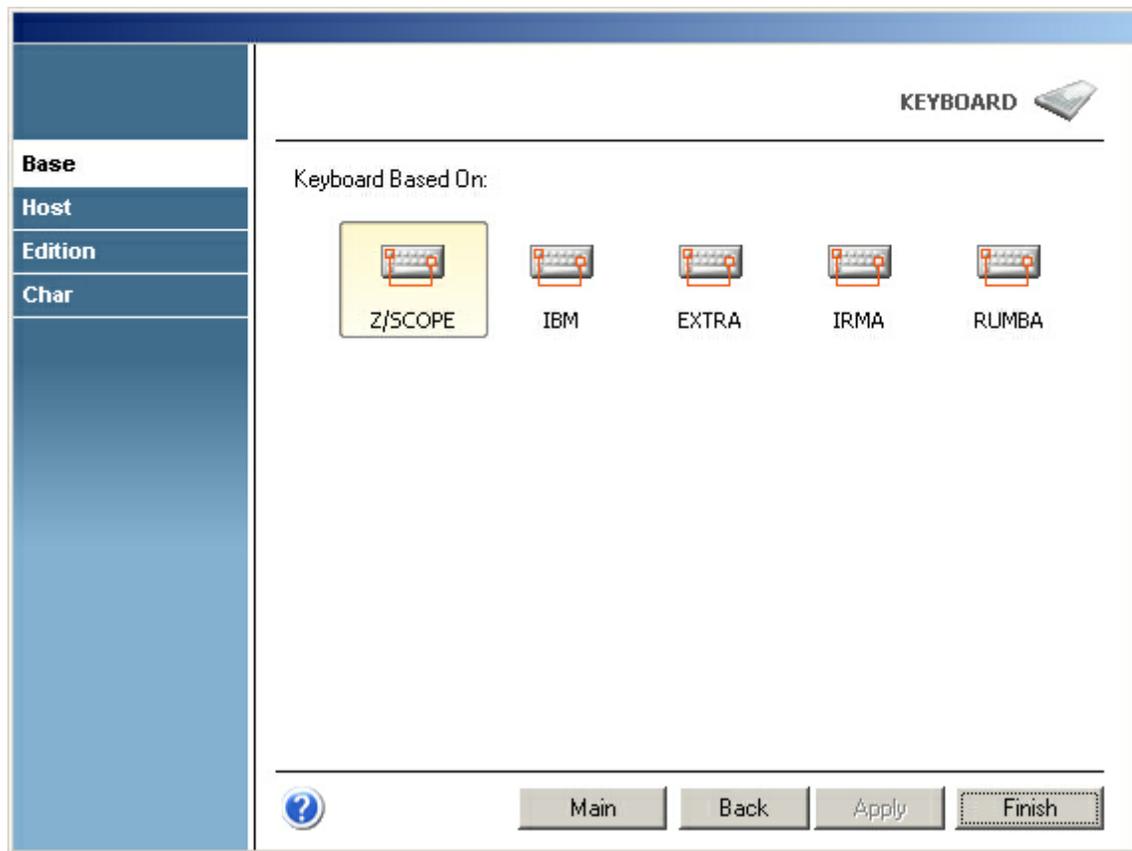


For more information about Keyboard settings, consult the following sections:

- [Base](#)
- [Host](#)
- [Edition](#)
- [Char](#)

4.8.1 Base

In the 'Base' tab you will find the following parameters:



Keyboard Based On

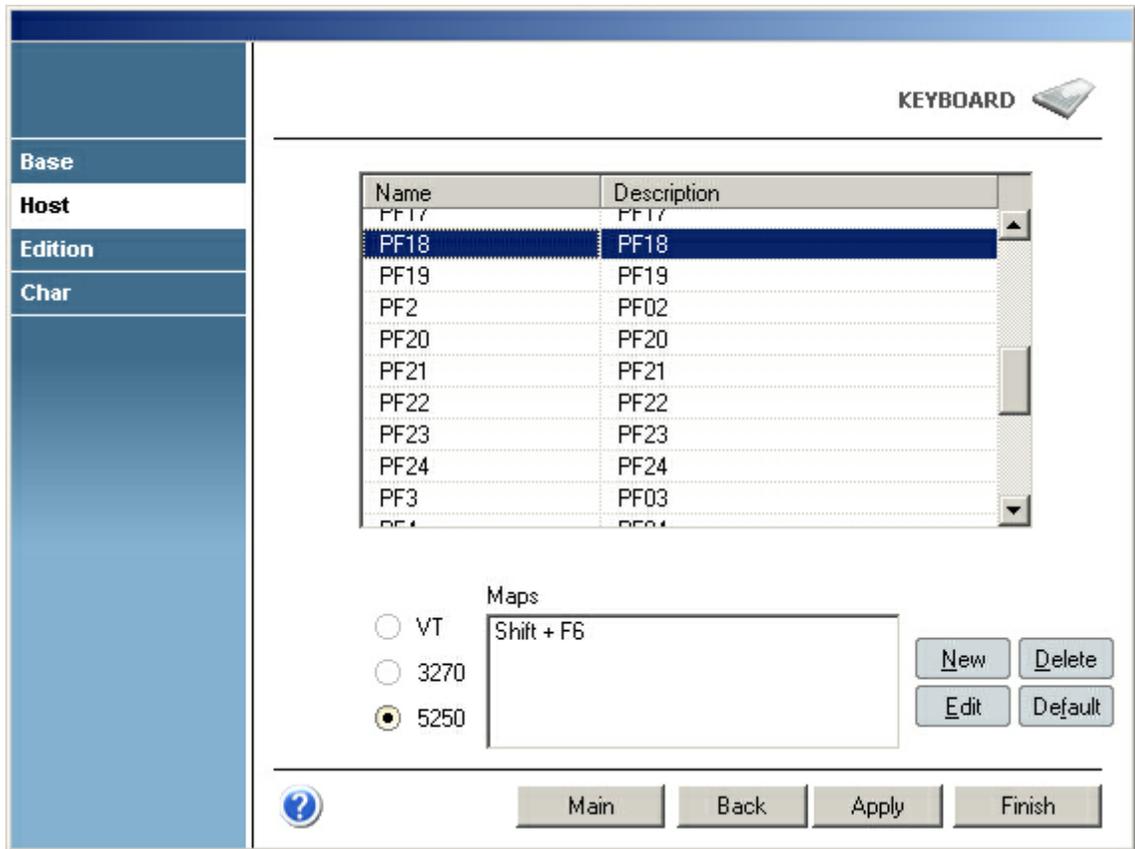
Select the keyboard layout that best matches the one you have.

To select the most appropriate default keyboard, visit the following sections:

- [z/Scope Keyboard Map](#)
- [IBM Keyboard Map](#)
- [EXTRA Keyboard Map](#)
- [IRMA Keyboard Map](#)
- [RUMBA Keyboard Map](#)

4.8.2 Host

In the 'Host' tab you will find the following parameters:

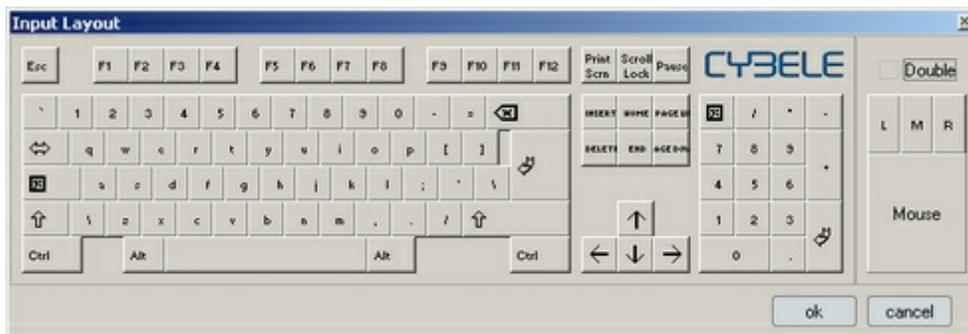


Grid

The grid contains a list of all host functions available for mapping. Those that appear in **red** mean their default mapping has been modified.

Maps

This list contains all keys mapped to the selected function. To map a new key to the selected function, click on the 'New' button to display the on-screen keyboard where you can easily choose a key combination.



Click 'Ok' when done. The selected key/s will be added to the list.

If you want to change previously assigned keys, select the key you want to modify from the list and then click on the 'Edit' button. Use the on-screen keyboard to select a new key and then click 'OK'.

To delete previously mapped keys, select a key from the list and then click on the 'Delete' Button.

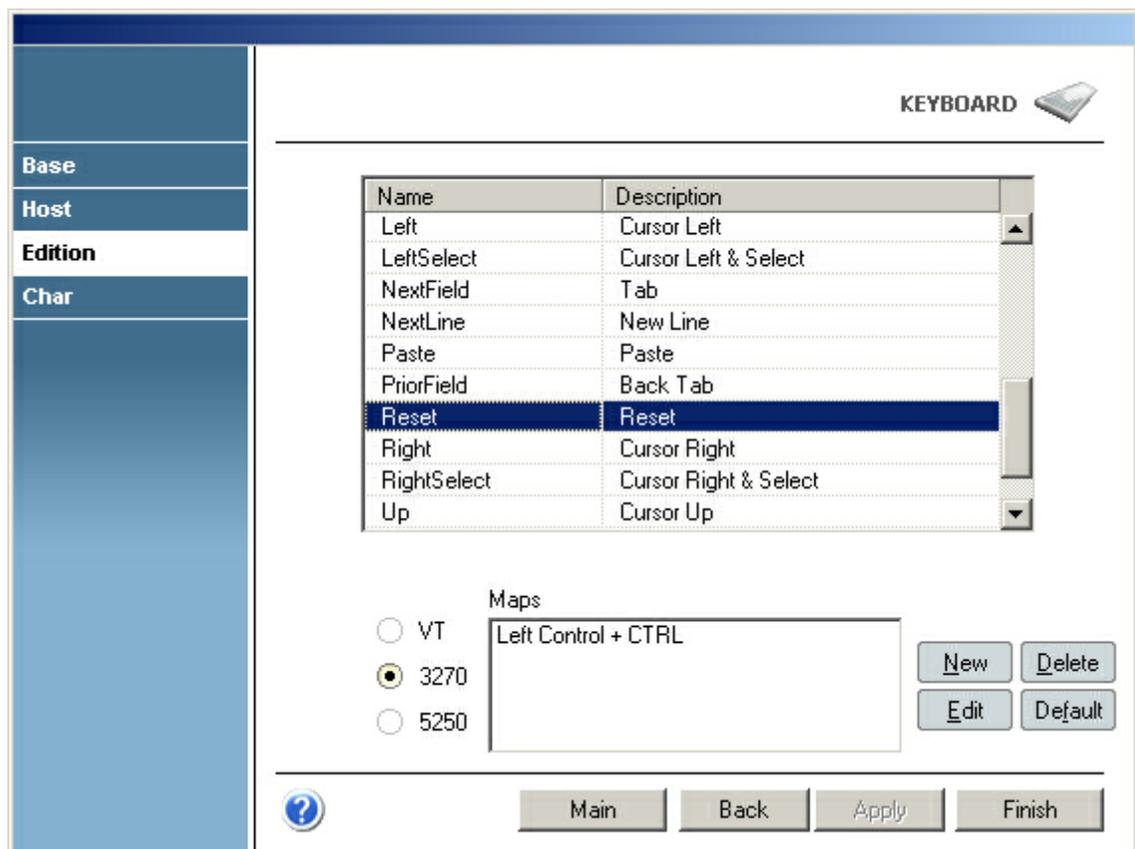
To restore the default mapping for the selected function, click on the 'Default' button.

3270/5250/VT filter

You can specify different maps for each host type.

4.8.3 Edition

In the 'Edition' tab you will find the following parameters:



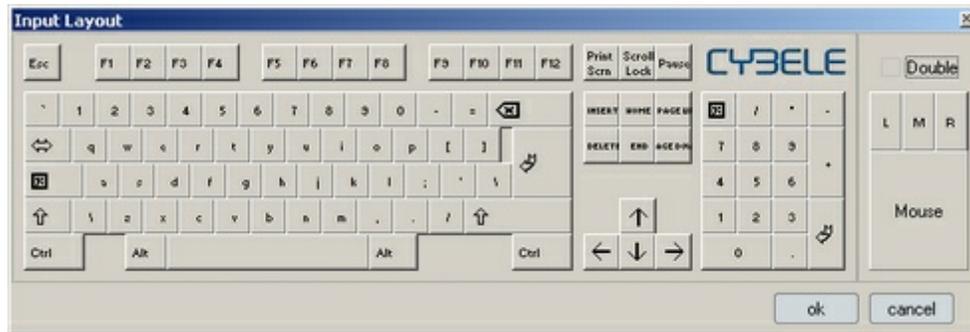
Grid

The grid contains a list of all edition functions available for mapping. Those that appear in red mean their default mapping has been modified.

Maps

This list contains all keys mapped to the selected function. To map a new key to the

selected function, click on the 'New' button to display the on-screen keyboard where you can easily choose a key combination.



Click 'Ok' when done. The selected key/s will be added to the list.

If you want to change previously assigned keys, select the key you want to modify from the list and then click on the 'Edit' button. Use the on-screen keyboard to select a new key and then click 'Ok'.

To delete previously mapped keys, select a key from the list and then click on the 'Delete' Button.

To restore the default mapping for the selected function, click on the 'Default' button.

3270/5250/VT filter

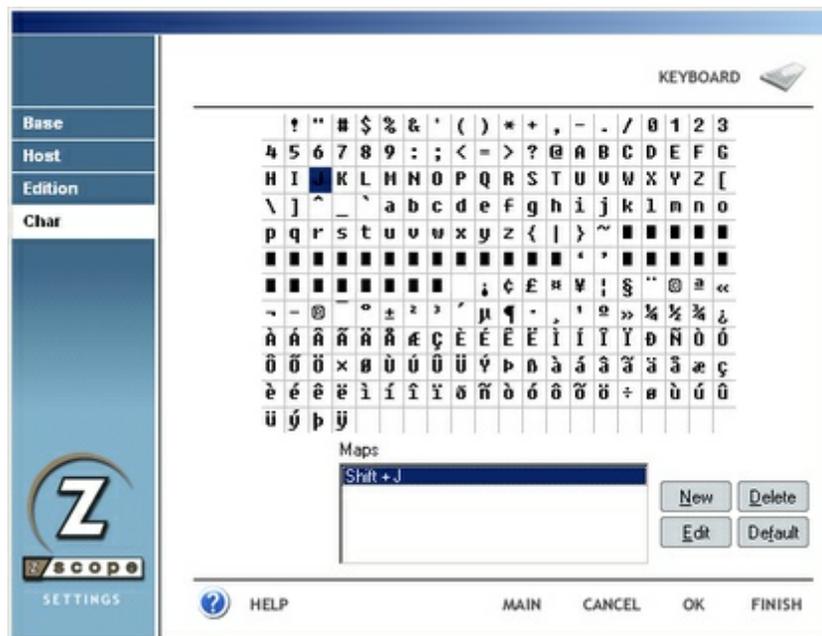
You can specify different maps for each host type.

Related Topics

[Default Keyboard Mapping](#)

4.8.4 Char

In the 'Char' tab you will find the following parameters:

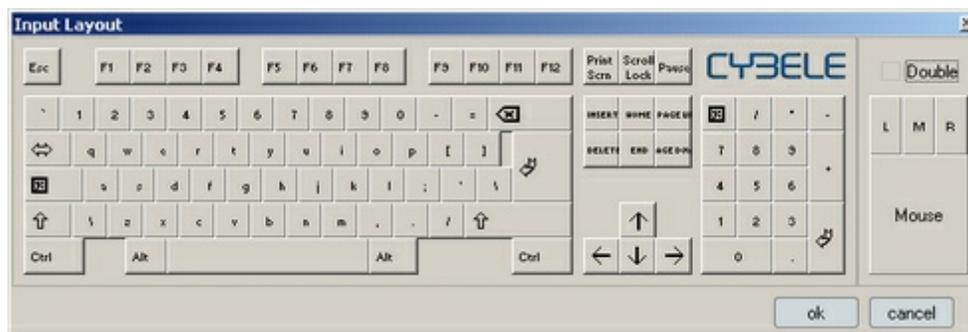


Character Grid

This is the entire character set. Choose the character you want to re-map.

Maps

This list contains all keys mapped to the selected character. To map a new key to the selected character, click on the 'New' button to display the on-screen keyboard where you can easily choose a key combination.



Click 'Ok' when done. The selected key/s will be added to the list.

If you want to change previously mapped keys, select the key you want to modify from the list and then click on the 'Edit' button. Use the on-screen keyboard to select a new key and then click 'Ok'.

To delete previously mapped keys, select a key from the list and then click on the 'Delete' Button.

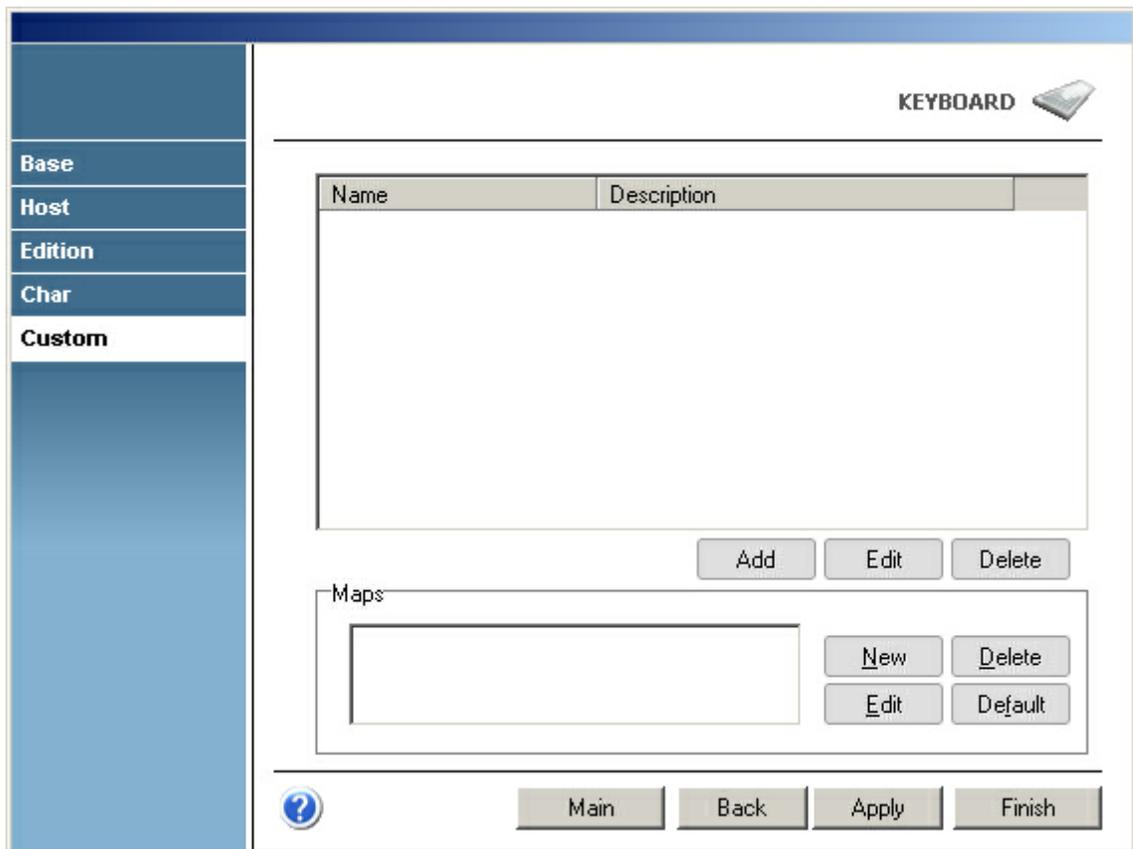
To restore the default mapping for the selected key, click on the 'Default' button.

Related Topics

Character Conversion Tables

4.8.5 Custom

In the 'Custom' tab you will find the following parameters:



Add

Press this button to add a new custom function. It will display the Keyboard Function dialog, which is explained below.

Edit

Select a function from the list and press this button to change any of the options for the function except its name.

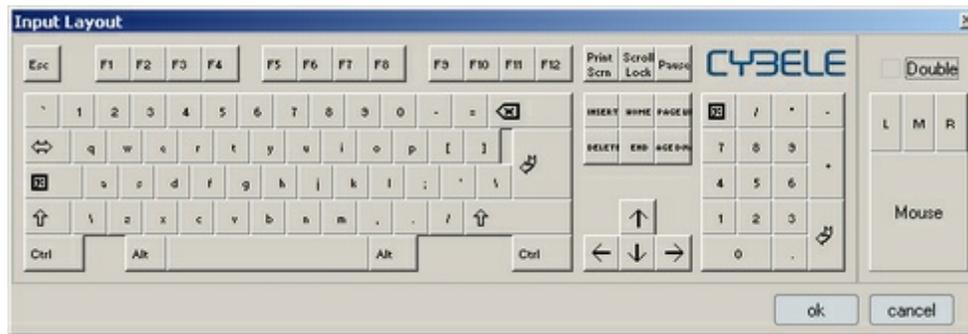
Delete

Select a function from the list and press this button to delete it from the custom functions list.

Maps

New

Select a function from the list and press this button to add a map for the selected function. It will display the Input Layout dialog box where you can easily choose a key combination.



Click 'Ok' when done. The selected key/s will be added to the list.

Delete

Select a function from the list and then a map for the selected function and press this button to delete the map.

Edit

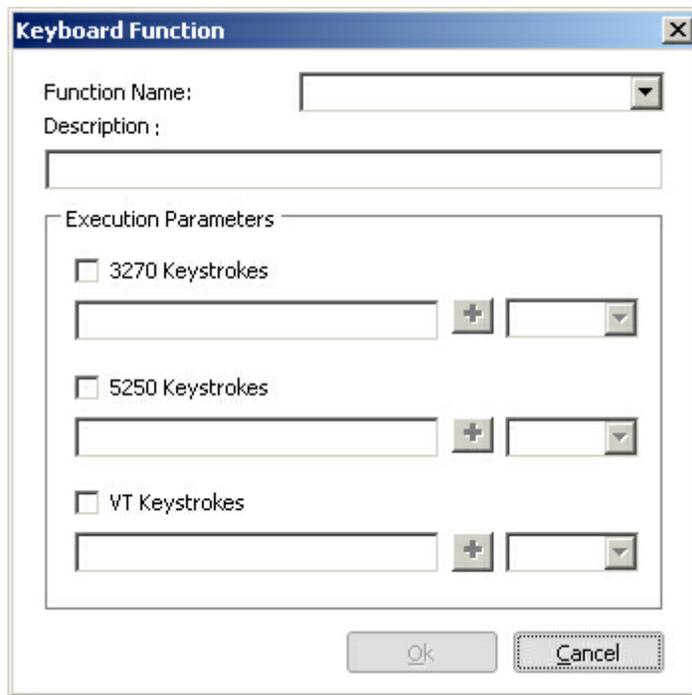
Select a function from the list and then a map for the selected function and press this button to change the map for a different one. The Input Layout dialog box will display and the map you select will replace the previous one.

Default

Select a function from the list and then a map for the selected function and press this button to change the map for the default one. In the Custom tab, the default map for all the functions is none, so pressing this button will delete all maps for the selected function.

Keyboard Function Dialog:

This dialog box is invoked by the 'Add' or 'Edit' buttons in the custom tab. If invoked from the 'Add' button, all the fields will appear blank and you can use it to add a new custom function. If invoked from the 'Edit' button, the fields will be completed with the information of the selected function and you can use it to change this information. You will find the following parameters:



Function Name

Enter here the function name. You can type a new name, or choose existing functions from the combobox.

Description

Enter a description for the custom function.

Execution parameters

Mark the hosts for which you want this function activated (3270, 5250, VT). Under each of them, type the series of keystrokes you want the system to send for each one. The combobox in the right holds a list of common keyboard functions, that you can add using the '+' button.

Ok

Press this button to Add the new function or confirm changes to the edition of an existing function.

Cancel

Press this button to discard the whole procedure.

4.8.6 Keyboards

4.8.6.1 z/Scope Keyboard Map

3270 Keyboard

Aids	Mapped keys
PF01 .. PF12	F1 .. F12

PF13 .. PF24	Shift + F1 .. Shift + F12
PA01 .. PA10	Left Control + F1 .. Left Control + F10
Attention	Left Control + Shift + A
Clear	Pause Shift + Pause Left Ctrl + Shift + Z
Enter	Enter Shift + Enter
Erase Input	Left Alt + End
Reset	Left Control Left Control + R
System Request	Left Alt + PrintScreen Left Alt + Multiply
Test Request	Left Alt + Pause

 If your keyboard does not distinguish keys then left keys are equivalent to right keys.

5250 Keyboard

Aids	Mapped keys
PF01 .. PF12	F1 .. F12
PF13 .. PF24	Shift + F1 .. Shift + F12
PA01 .. PA03	Left Control + F1 .. Left Control + F3
Attention	Escape
Clear	Pause Shift + Pause
Enter	Enter
Erase EOF	End
Erase EOL	Left Alt + Home
Erase Input	Left Alt + End
Field - (minus)	Minus Left Shift + Minus
Field + (plus)	Plus Shift + Plus
Help	Scroll Lock Shift + Scroll Lock
New Line	Shift + Enter
Reset	Left Control
System Request	Left Alt + Print Screen Left Alt + Multiply
Test Request	Left Alt + Pause

 If your keyboard does not distinguish keys then left keys are equivalent to right keys.

4.8.6.2 IBM Keyboard Map

IBM 3270 Keyboard Map

Function Key	Mapped keys
Attn	Esc
Back Space	Backspace
Back Tab	Left Shift + Tab Right Shift + Tab
Clear	Pause
Cursor Down	Down Num 2
Cursor Down & Select	Left Shift + Down Left Shift + Num 2 Right Shift + Down Right Shift + Num 2
Cursor Left	Left Num 4
Cursor Left & Select	Left Shift + Left Left Shift + Num 4 Right Shift + Left Right Shift + Num 4
Cursor Right	Right Num 6
Cursor Right & Select	Left Shift + Right Left Shift + Num 6 Right Shift + Right Right Shift + Num 6
Cursor Ruler	Left Control + Home Right Control + Home
Cursor Up	Up Num 8
Cursor Up & Select	Left Shift + Up Left Shift + Num 8 Right Shift + Up Right Shift + Num 8
Delete	Delete Num Del
Dup	Left Shift + Insert Right Shift + Insert
End	End Num 1
Enter	Num Enter Right Control + Right Ctrl
Erase Eof	Left Control + End Right Control + End
Erase Input	Left Alt + End Right Alt + End
Field Mark	Left Shift + Home Right Shift + Home

Home	Home Num 7
Insert	Num 0 Insert
New Line	Enter
PA1	Left Alt + Insert Right Alt + Insert
PA2	Left Alt + Home Right Alt + Home
PA3	Left Shift + Page Up Right Shift + Page Up
PF01	F1
PF02	F2
PF03	F3



If your keyboard does not distinguish keys then left keys are equivalent to right keys.

IBM 5250 Keyboard Map

Function Key	Mapped keys
Attn	Esc
Back Space	Backspace
Back Tab	Right Shift + Tab Left Shift + Tab
Cursor Down	Down Num 2
Cursor Down & Select	Left Shift + Down Left Shift + Num 2 Right Shift + Down Right Shift + Num 2
Cursor Left	Left Num 4
Cursor Left & Select	Left Shift + Left Left Shift + Num 4 Right Shift + Left Right Shift + Num 4
Cursor Right	Right Num 6
Cursor Right & Select	Left Shift + Right Left Shift + Num 6 Right Shift + Right Right Shift + Num 6
Cursor Ruler	Left Control + Home Right Control + Home
Cursor Up	Up Num 8
Cursor Up & Select	Left Shift + Up Left Shift + Num 8

	Right Shift + Up Right Shift + Num 8
Delete	Delete Num Del
Dup	Left Shift + Insert Right Shift + Insert
End	End
Enter	Right Control + Right Ctrl
Erase Eof	Right Control + End Left Control + End
Erase Input	Left Alt + End Right Alt + End
Field -	Num - Left Shift + Num - Right Shift + Num - Left Alt + Num - Right Alt + Num -
Field +	Num Enter Left Shift + Num Enter Right Shift + Num Enter Num + Right Shift + Num + Left Shift + Num +
Field Mark	Left Shift + Home Right Shift + Home
Help	Left Alt + F1 Right Alt + F1
Home	Home Num 7
Home	Home Num 7
Insert	Insert Num 0
New Line	Enter
PA1	Left Alt + Insert Right Alt + Insert
PA2	Left Alt + Home Right Alt + Home
PA3	Left Shift + Page Up Right Shift + Page Up



If your keyboard does not distinguish keys then left keys are equivalent to right keys.

4.8.6.3 EXTRA Keyboard Map

EXTRA 3270 Keyboard Map

Function Key	Mapped keys
--------------	-------------

Attn	Left Shift + Esc Right Shift + Esc
Back Space	Backspace
Back Tab	Left Shift + Tab Right Shift + Tab
Clear	Pause
Cursor Down	Down Num 2
Cursor Down & Select	Left Shift + Down Left Shift + Num 2 Right Shift + Down Right Shift + Num 2
Cursor Left	Left Num 4
Cursor Left & Select	Left Shift + Left Left Shift + Num 4 Right Shift + Left Right Shift + Num 4
Cursor Right	Right Num 6
Cursor Right & Select	Left Shift + Right Left Shift + Num 6 Right Shift + Right Right Shift + Num 6
Cursor Ruler	Left Control + Num 7 Right Control + Num 7
Cursor Up	Up Num 8
Cursor Up & Select	Left Shift + Up Left Shift + Num 8 Right Shift + Up Right Shift + Num 8
Delete	Delete Num Del
Dup	Left Shift + Insert Right Shift + Insert
End	End
Enter	Num Enter Enter
Erase Eof	Num 1 End
Erase Input	Left Alt + End Right Alt + End
Field Mark	Left Shift + Home Right Shift + Home
Home	Home Num 7
Insert	Insert Num 0
New Line	Right Control + Right Ctrl

PA1	Num 9 Page Up
PA2	Num 3 Page Down
PA3	Left Shift + Page Down Right Shift + Page Down
PF01	F1
PF02	F2
PF03	F3



If your keyboard does not distinguish keys then left keys are equivalent to right keys.

EXTRA 5250 Keyboard Map

Function Key	Mapped keys
Attn	Left Shift + Esc Right Shift + Esc
Back Space	Backspace
Back Tab	Right Shift + Tab Left Shift + Tab
Cursor Down	Down Num 2
Cursor Down & Select	Left Shift + Down Left Shift + Num 2 Right Shift + Down Right Shift + Num 2
Cursor Left	Left Num 4
Cursor Left & Select	Left Shift + Left Left Shift + Num 4 Right Shift + Left Right Shift + Num 4
Cursor Right	Right Num 6
Cursor Right & Select	Left Shift + Right Left Shift + Num 6 Right Shift + Right Right Shift + Num 6
Cursor Ruler	Left Control + Num 7 Right Control + Num 7
Cursor Up	Up Num 8
Cursor Up & Select	Left Shift + Up Left Shift + Num 8 Right Shift + Up Right Shift + Num 8
Delete	Delete Num Del
Dup	Left Shift + Insert

	Right Shift + Insert
End	End
Enter	Enter Num Enter
Erase Eof	Num 1 End
Erase Input	Left Alt + End Right Alt + End
Field -	Num -
Field +	Num + Left Shift + Num + Right Shift + Num +
Field Mark	Left Shift + Home Right Shift + Home
Help	Left Alt + Pause Right Alt + Pause
Home	Num 7 Home
Home	Home Num 7
Insert	Insert Num 0
New Line	Right Control + Right Ctrl
PA1	Left Alt + Insert Right Alt + Insert
PA2	Left Alt + Home Right Alt + Home
PA3	Left Shift + Page Down Right Shift + Page Down Left Alt + Page Up Right Alt + Page Up



If your keyboard does not distinguish keys then left keys are equivalent to right keys.

4.8.6.4 IRMA Keyboard Map

IRMA 3270 Keyboard Map

Function Key	Mapped Keys
Attn	F1
Back Space	Backspace
Back Tab	Left Shift + Tab Right Shift + Tab
Clear	F2
Cursor Down	Down Num 2
Cursor Down & Select	Left Shift + Down

	Left Shift + Num 2 Right Shift + Down Right Shift + Num 2
Cursor Left	Left Num 4
Cursor Left & Select	Left Shift + Left Left Shift + Num 4 Right Shift + Left Right Shift + Num 4
Cursor Right	Right Num 6
Cursor Right & Select	Left Shift + Right Left Shift + Num 6 Right Shift + Right Right Shift + Num 6
Cursor Ruler	Left Control + Home Right Control + Home
Cursor Up	Up Num 8
Cursor Up & Select	Left Shift + Up Left Shift + Num 8 Right Shift + Up Right Shift + Num 8
Delete	Delete Num Del
Dup	Left Control + G Right Control + G
End	End Num 1
Enter	Enter Num Enter
Erase Eof	F6
Erase Input	F4
Field Mark	Left Control + H Right Control + H
Home	Home Num 7
Insert	Num 0 Insert
New Line	Left Control + Enter Right Control + Enter
PA1	Left Control + J Right Control + J
PA2	Left Control + K Right Control + K
PA3	Left Control + L Right Control + L
PF01	Left Alt + 1 Right Alt + 1
PF02	Left Alt + 2

	Right Alt + 2
PF03	Left Alt + 3 Right Alt + 3

 If your keyboard does not distinguish keys then left keys are equivalent to right keys.

IRMA 5250 Keyboard Map

Function Key	Mapped keys
Attn	Esc
Back Space	Backspace
Back Tab	Right Shift + Tab Left Shift + Tab
Cursor Down	Down Num 2
Cursor Down & Select	Left Shift + Down Left Shift + Num 2 Right Shift + Down Right Shift + Num 2
Cursor Left	Left Num 4
Cursor Left & Select	Left Shift + Left Left Shift + Num 4 Right Shift + Left Right Shift + Num 4
Cursor Right	Right Num 6
Cursor Right & Select	Left Shift + Right Left Shift + Num 6 Right Shift + Right Right Shift + Num 6
Cursor Ruler	Left Control + Home Right Control + Home
Cursor Up	Up Num 8
Cursor Up & Select	Left Shift + Up Left Shift + Num 8 Right Shift + Up Right Shift + Num 8
Delete	Delete Num Del
Dup	Left Shift + Right Control + Insert Right Shift + Insert
End	End
Enter	Enter
Erase Eof	Left Control + End
Erase Input	Left Alt + End
Field -	Num -

	Left Shift + Num - Right Shift + Num - Left Alt + Num - Right Alt + Num -
Field +	Num Enter Left Shift + Num Enter Right Shift + Num Enter Num + Left Control + Num + Left Alt + Num + Right Control + Num + Right Alt + Num +
Field Mark	Left Shift + Home Right Shift + Home
Help	Left Alt + F1 Right Alt + F1
Home	Num 7 Home
Home	Home Num 7
Insert	Insert Num 0
New Line	Enter
PA1	Left Alt + Insert
PA2	Right Control + Left Alt + Home
PA3	Right Alt + Page Up Right Control + L



If your keyboard does not distinguish keys then left keys are equivalent to right keys.

4.8.6.5 RUMBA Keyboard Map

RUMBA 3270 Keyboard Map

Function Key	Mapped keys
Attn	Left Shift + Left Control + A Right Shift + Left Control + A Left Shift + Right Control + A Right Shift + Right Control + A
Back Space	Left Shift + Backspace Right Shift + Backspace Backspace
Back Tab	Left Shift + Tab Right Shift + Tab
Clear	Left Shift + Left Control + Z Right Shift + Left Control + Z Right Shift + Right Control + Z Left Shift + Right Control + Z
Cursor Down	Down

	Num 2
Cursor Down & Select	Left Shift + Down Left Shift + Num 2 Right Shift + Down Right Shift + Num 2
Cursor Left	Left Num 4
Cursor Left & Select	Left Shift + Left Left Shift + Num 4 Right Shift + Left Right Shift + Num 4
Cursor Right	Right Num 6
Cursor Right & Select	Left Shift + Right Left Shift + Num 6 Right Shift + Right Right Shift + Num 6
Cursor Ruler	Left Control + Home
Cursor Up	Up Num 8
Cursor Up & Select	Left Shift + Up Left Shift + Num 8 Right Shift + Up Right Shift + Num 8
Delete	Delete Num Del
Dup	Left Control + D Right Control + D
End	End
Enter	Enter Num Enter
Erase Eof	End
Erase Input	Left Control + T Right Control + T
Field Mark	Left Shift + Home Right Shift + Home
Home	Home Num 7
Insert	Insert Num 0
New Line	Left Control + Enter
PA1	Left Control + F1 Right Control + F1
PA2	Left Control + F2 Right Control + F2
PA3	Left Control + F3 Right Control + F3
PF01	F1
PF02	F2
PF03	F3

 If your keyboard does not distinguish keys then left keys are equivalent to right keys.

RUMBA 5250 Keyboard Map

Function Key	Mapped keys
Attn	Esc
Back Space	Backspace Left Shift + Backspace Right Shift + Backspace
Back Tab	Right Shift + Tab Left Shift + Tab
Cursor Down	Down Num 2
Cursor Down & Select	Left Shift + Down Left Shift + Num 2 Right Shift + Down Right Shift + Num 2
Cursor Left	Left Num 4
Cursor Left & Select	Left Shift + Left Left Shift + Num 4 Right Shift + Left Right Shift + Num 4
Cursor Right	Right Num 6
Cursor Right & Select	Left Shift + Right Left Shift + Num 6 Right Shift + Right Right Shift + Num 6
Cursor Ruler	Left Alt + Page Down Right Alt + Page Down
Cursor Up	Up Num 8
Cursor Up & Select	Left Shift + Up Left Shift + Num 8 Right Shift + Up Right Shift + Num 8
Delete	Delete Num Del
Dup	Left Shift + Insert Right Shift + Insert
End	End
Enter	Enter
Erase Eof	Left Control + End Right Control + End
Erase Input	Left Alt + End Right Alt + End
Field -	Num -

	Left Shift + Num - Right Shift + Num -
Field +	Right Control + Right Ctrl Num +
Field Mark	Left Shift + Home Right Shift + Home
Help	Scroll Lock
Home	Home Num 7
Insert	Insert Num 0
New Line	Left Shift + Enter Right Shift + Enter
PA1	Left Control + F1 Right Control + F1
PA2	Left Control + F2 Right Control + F2
PA3	Left Control + F3 Right Control + F3

 If your keyboard does not distinguish keys then left keys are equivalent to right keys.

4.9 Environment

The Environment is managed from the 'Environment' icon in the 'Settings' dialog.

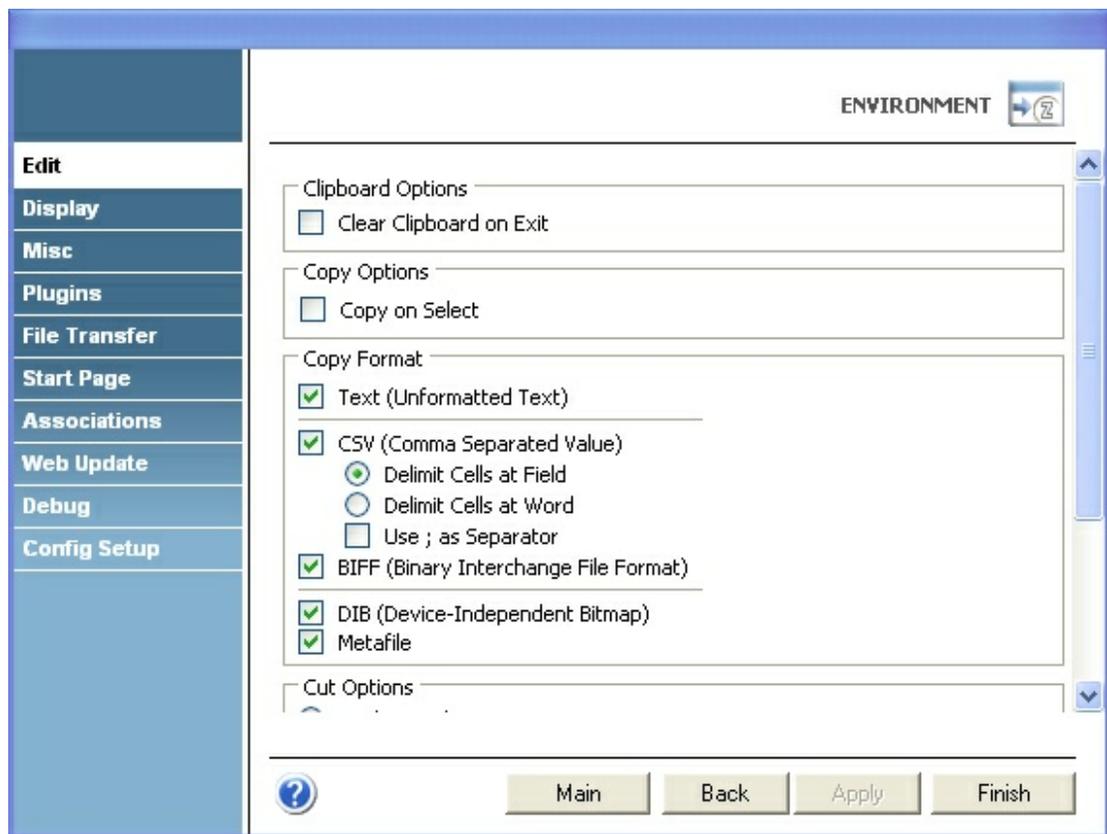


For more information about the Environment settings available, consult the following sections:

- [Edit](#)
- [Display](#)
- [Misc](#)
- [Plugins](#)
- [File Transfer](#)
- [Start Page](#)
- [Associations](#)
- [Debug](#)
- [Config Setup](#)

4.9.1 Edit

In the 'Edit' tab you will find the following parameters:



Clipboard Options

Clear clipboard on exit

Check this option to clear Windows' clipboard when closing z/Scope. This is useful for security reasons when you have to place confidential information into the clipboard during z/Scope sessions.

Copy Options

Copy on select

Check this option to automatically copy the selection you make into the clipboard.

Copy Format

Text (unformatted text)

Check this option if you want to copy text into the clipboard unformatted.

CSV (comma separated value)

Check this option if you want to copy text into the clipboard as comma-separated values. You can also specify if cells are to be delimited as fields or at words. Instead of a comma (,), you can use the semi-colon.

BIFF (binary interchange file format)

Check this option if you want to copy text into the clipboard in BIFF format.

DIB (device-independent bitmap)

Check this option if you want to copy text into the clipboard in DIB format.

Metafile

Check this option if you want to copy text into the clipboard in Metafile format.

Cut Options

Here you can choose whether spaces or nulls will be placed when you perform a cut operation.

Paste Options

Skip on protected fields

Check this option if you want paste operation to skip on protected fields.

Move cursor after paste

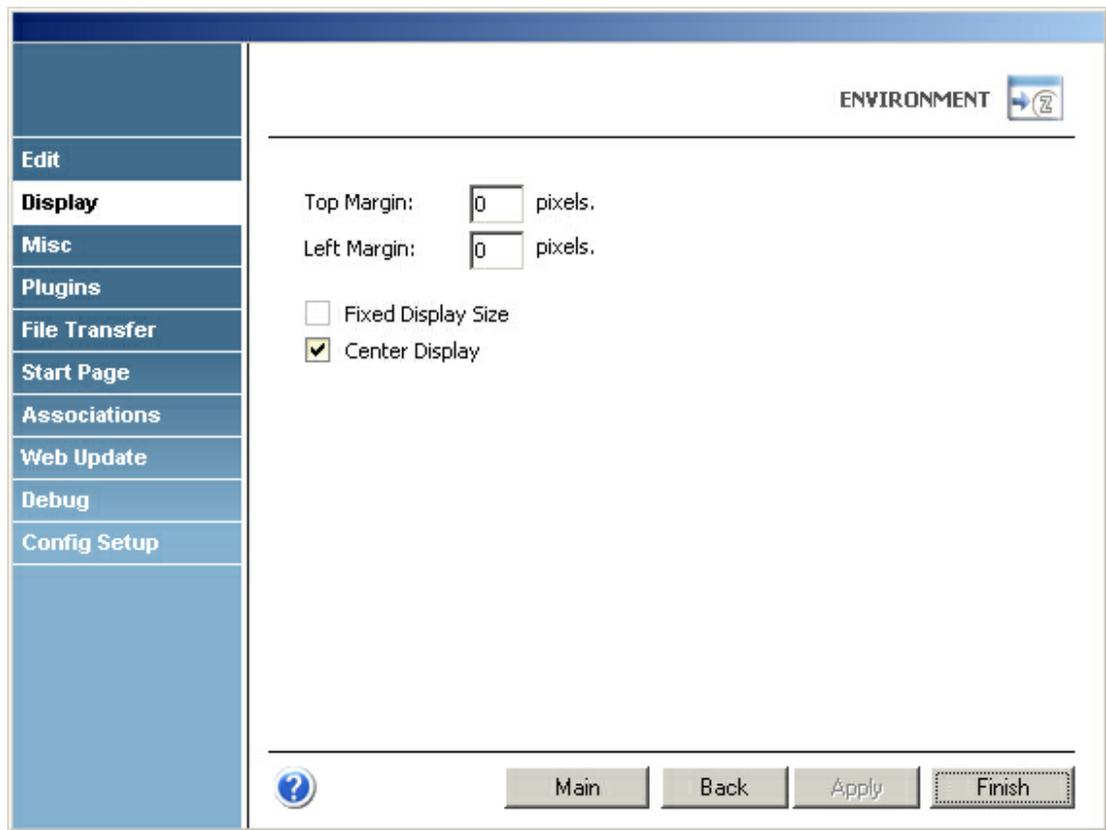
Check this option if you want the cursor to move automatically to the end of the pasted end after pasting.

Replace TABs with

Check this option if you want to replace TAB chars with text when pasting. Specify the text in the field below.

4.9.2 Display

In the 'Display' tab you will find the following parameters:



Top and Left Margins

Here you can specify the size (in pixels) of the top and left margins. Default value for both margins is 0 (no margins).

Fixed Display Size

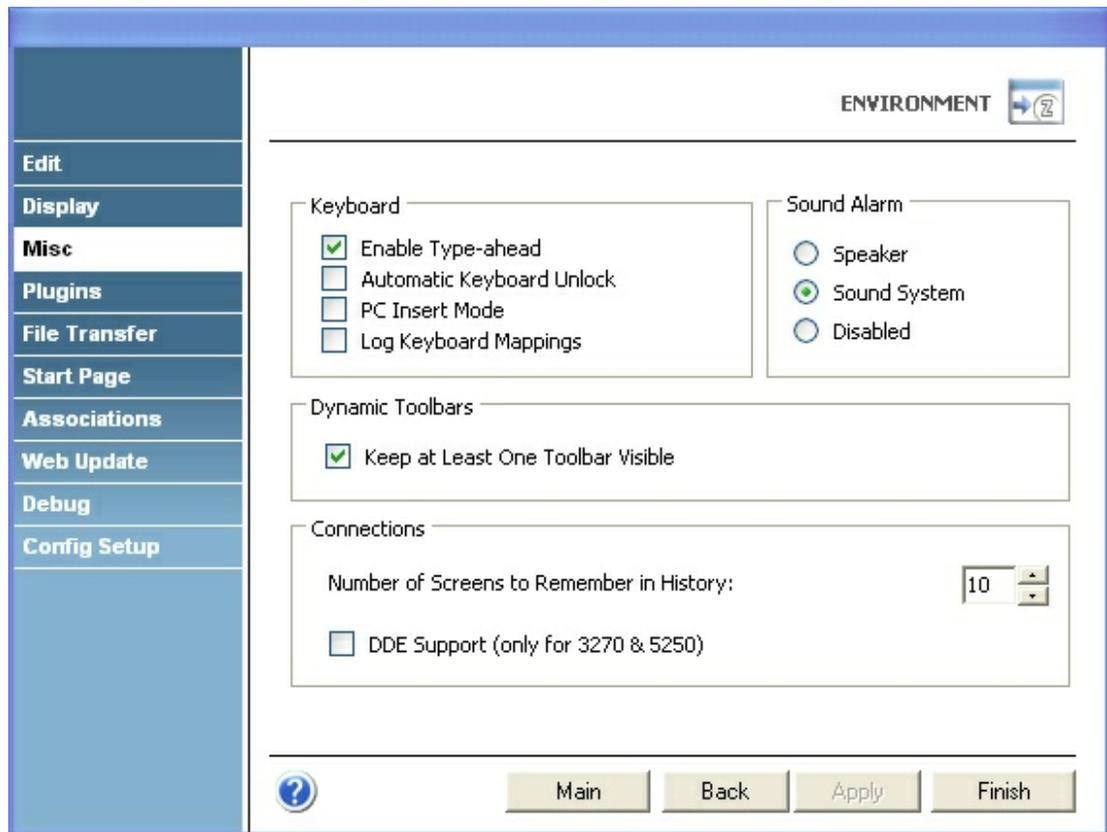
Check this option if you want character and line spacings to remain fixed when resizing the emulation.

Center Display

Check this option if you want the emulation display to be centered.

4.9.3 Misc

In the 'Misc' tab you will find the following parameters:



Keyboard

Enable Type-Ahead

Check this option if you want keystrokes to be sent to a buffer when the screen is still loading so the characters typed during that interval are sent when the screen is once again ready for input.

Automatic Keyboard Unlock

Check this option if you want the keyboard to be automatically unlocked after typing in a protected area of the screen.

PC Insert Mode

Check this option if you want to work in the Insert mode instead of the Overwrite mode when typing.

Sound Alarm

Specify if systems beeps are played through the PC speaker or the PC's sound card. Select 'Disabled' if you don't want any sounds to be played at all.

Dynamic Toolbars

Keep at least one toolbar visible

Check this option to specify that you want at least one toolbar to be visible at the top of the Emulation Display.

Connections

Number of Screens to remember in History

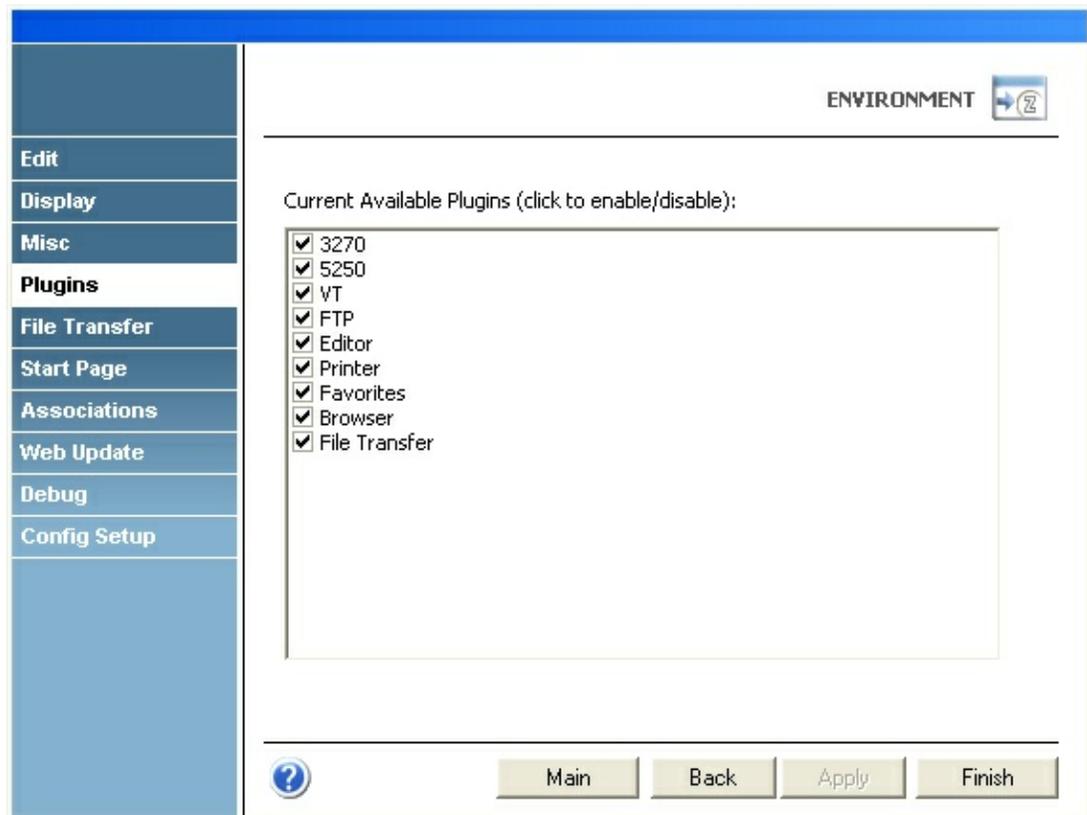
In this field you can specify the number of screens that will be kept in the [Screen History](#).

DDE Support

This checkbox enables Dynamic Data Exchange. This option works only on TE3270 and TE5250 connections.

4.9.4 Plugins

In the 'Plugins' tab you will find a list of the available plugins, which you can check or uncheck to customize the use of z/Scope. After making your selection you will need to restart the application for the changes to take effect.



Important:

The FTP function is not available for Express editions and might be not included on your License.

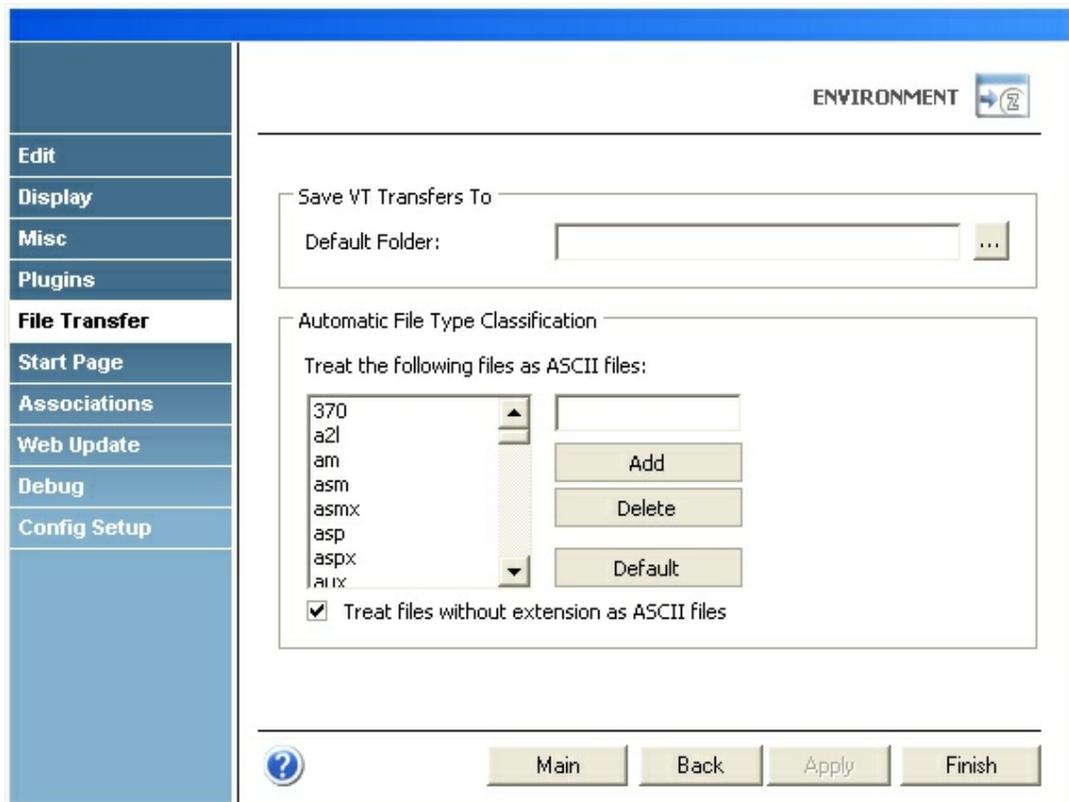
Also note that the Editor works only on z/Scope Workbench.

[See our z/Scope Edition Comparison online.](#)

To learn more about plugins and their functionality, read on: [Appendix: Plugins](#)

4.9.5 File Transfer

In the 'File Transfer' tab you can specify the default folder to save VT file transfers in your computer. You can also decide to treat some file types as ASCII files.



4.9.6 Start Page

In the 'Start Page' tab you will find the following parameters:

The screenshot shows the 'ENVIRONMENT' settings window for z/Scope. On the left is a navigation pane with options: Edit, Display, Misc, Plugins, File Transfer, Start Page (selected), Associations, Web Update, Debug, and Config Setup. The main area is divided into three sections: 'Start Page', 'Gradient Areas', and 'Visibility'. In the 'Start Page' section, the 'Default Category' dropdown is set to 'CONNECTIONS', the 'Feed' dropdown is empty, and the 'Use Custom URL' text box contains 'http://www.cybelesoft.com/'. The 'Gradient Areas' section has 'Top Color' and 'Bottom Color' dropdowns set to 'Custom...' and 'Font Color' set to 'Default'. The 'Visibility' section has three checked checkboxes: 'Show On Start', 'Show Favorites Panel', and 'Show Categories Panel'. At the bottom are buttons for 'Main', 'Back', 'Apply', and 'Finish', along with a help icon.

Start Page

Default Category:

Mark this option if you would like to see a feed or the connections list in the start page every time you open z/Scope.

Feed:

If you have chosen to see a feed on the start page, select the feed you want to see from the available list.

Use Custom URL:

Mark this option if you would like to see a web site in the start page every time you open z/Scope, and enter the URL.

Gradient Areas

Top Color:

Choose the upper color that will conform the gradient for the panel of the start page.

Bottom Color:

Choose the bottom color that will conform the gradient for the panel of the start page.

Font Color:

Choose a font color for the text. This text will appear above the gradient area.

Visibility

Show Start:

Mark this option if you want to have the start page visible by default everytime you run z/Scope.

Show Favorites Panel:

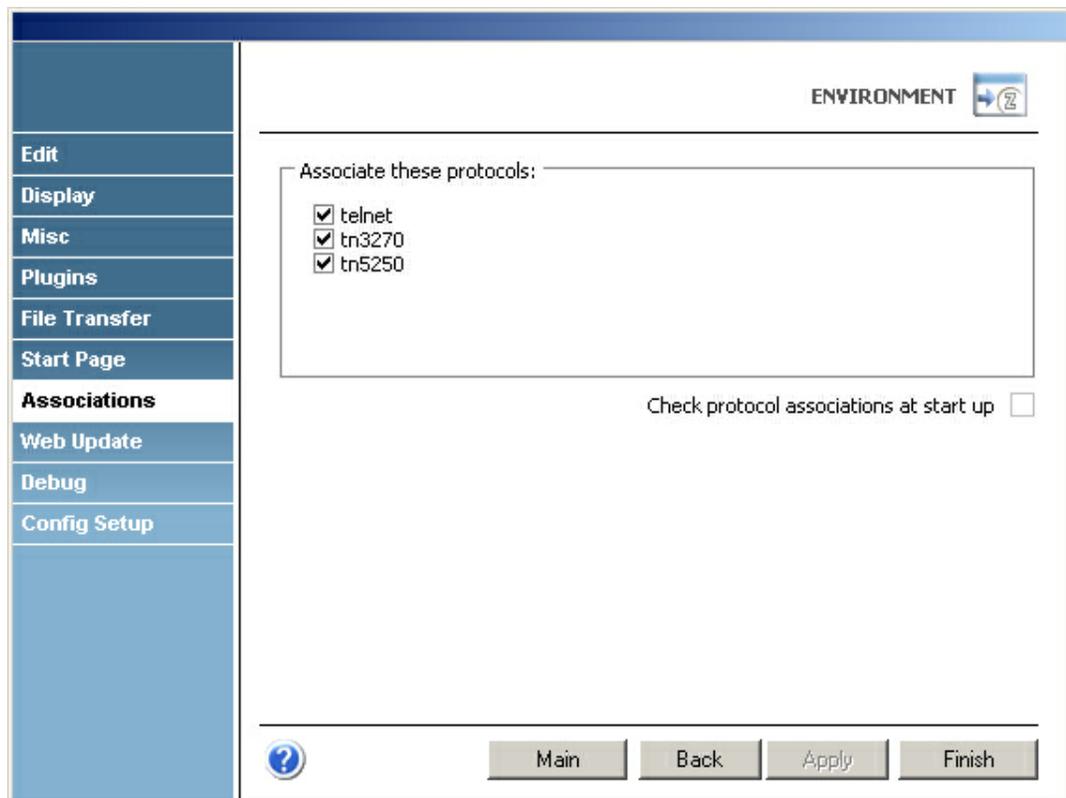
Mark this option if you want to see the Favorites Panel in the Start Page.

Show Categories

Mark this option if you want to see the Categories Panel in the Start Page. The Categories Panel has a direct link to the list of connections.

4.9.7 Associations

In the 'Associations' tab you will find the following parameters:



Associate these protocols:

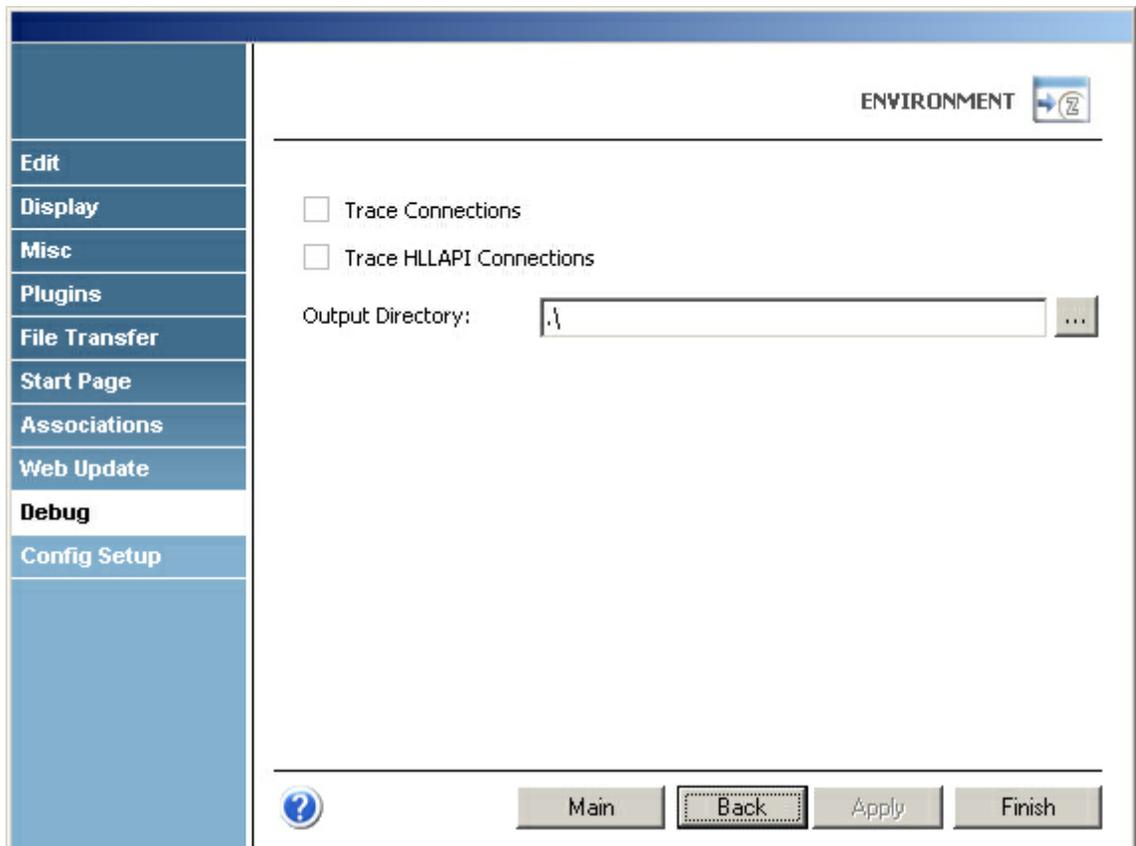
Select which protocols to associate directly with z/Scope, to make it your default application.

Check protocol associations at start up:

Mark this option if you would like z/Scope to check on start up whether it is your default application for the available protocols.

4.9.8 Debug

In the 'Debug' tab you will find the following parameters:



Trace Connections

Check this option if you want to keep track of the connection's activity in a dump file that can be used later to trace errors.

Trace HLLAPI Connections

Check this option if you want to keep track of HLLAPI connections' activity in a dump file that can be used later to trace errors.

Output Directory

Here you can specify the directory where the trace files will be placed.

4.9.9 Config Setup

In the 'Config Setup' tab you will find the following parameters:

The screenshot shows the 'Config Setup' dialog box. The sidebar on the left lists various configuration categories, with 'Config Setup' currently selected. The main window is titled 'ENVIRONMENT' and features a 'Configuration Path' text box containing the path '\$APPDATA\Cybele Software\zScope6\'. Below this, there is an unchecked checkbox labeled 'Use Remote Configuration Server' and an empty 'URL:' text box. At the bottom of the dialog, there are five buttons: a help icon (?), 'Main', 'Back', 'Apply', and 'Finish'.

Configuration Path

Enter here the path where you want to store your configuration files. You can enter different paths per Windows users. If you want to share your configuration files between Windows users, enter the same path.

Everytime you change your path, you will cease to use your current configuration files which will be kept in the previous path. At the same time, you will begin to use the files that are in the new path you have selected, and if this is the first you select that path, you will need to create all the configuration again.

Use Remote Configuration Server

Check this option and enter the URL where the remote files will be saved. This will allow all your z/Scope users to load the same configuration files.

5 Programming Reference

In this section you will find the following topics:

- [Scripting Objects](#)
- [HLLAPI Reference](#)

- [OHIO Reference](#)
- [DDE Reference](#)

5.1 Scripting Objects

To develop your scripts, you can use the Display Object:

- [Display Object](#)

5.1.1 Display Object

This object acts as interface with the host emulation screen. For more information about how to write scripts using the Display Object, consult the following sections:

- [Properties](#)
- [Methods](#)

5.1.1.1 Properties

This section covers the Display object's properties. For information on any particular property, go to the following links:

- [Cols](#)
- [Rows](#)
- [CursorPos](#)
- [EditFields](#)
- [HostFields](#)
- [HTML](#)
- [ScreenId](#)
- [ScreenRow](#)
- [Text](#)

5.1.1.1.1 Cols

The Cols property retrieves the total number of columns supported by the current terminal type.

Syntax

[*integer* =] Display.Cols

Remarks

Possible values for this property include for example 80 and 132, depending on the terminal type you are working with.

Related Topics

- [Rows Property](#)

5.1.1.1.2 Rows

The Rows property retrieves the total number of rows supported by the current terminal type.

Syntax

[*Integer* =] Display.Rows

Remarks

Possible values for this property include for example 24, 32 and 43, depending on the terminal type you are working with.

Related Topics

- [Cols Property](#)

5.1.1.1.3 CursorPos

The CursorPos property retrieves and sets the cursor's current screen position.

Syntax

[*integer* =] Display.CursorPos [= *cursor position*]

Remarks

The *cursor position* parameter is an integer ranging from 1 up to the total number of rows, multiplied by the total number of columns available for the terminal type specified.

The cursor position parameter can be calculated by multiplying the total number of columns by the cursor's row number minus one, and then adding the cursor's column number:

Cursor position = Total columns * (cursor's row number - 1) + cursor's column number

5.1.1.1.4 EditFields

`EditFields` is a collection that contains all the editable (unprotected) fields in the current screen.

Syntax

```
[string =] Display.EditFields(index|name) = [string]
```

Remarks

Fields in this collection can be accessed either by index or name. The name is constructed using its screen coordinates as follows: R[*row*]C[*column*]. For example, the field at row 4 column 30 in the screen can be referenced as "R4C30".

z/Scope automatically assigns screen fields' names based on the labels that identify them in the screen layout. In this way, in most cases, you can also reference a field in the screen using its label (for example "UserID"). Keep in mind that no blank spaces or characters outside the A-Z range are allowed for label references. For example, if a field in the screen is labeled "NEXT COMMAND:", it will be referenced as "NEXTCOMMAND".

If an invalid index value or field name is specified, a *Null* value will be returned.

Related Topics

- [HostFields Collection](#)

5.1.1.1.5 HostFields

`HostFields` is a collection that contains all the fields in the current screen.

Syntax

[*string* =] Display.[HostFields](#)(*index|name*)

Remarks

Fields in HostFields collection can be accessed either by index or name. The name is constructed using its screen coordinates as follows: R[*row*]C[*column*]. For example, the field at row 4 column 30 in the screen can be referenced as "R4C30".

z/Scope automatically assigns screen fields' names based on the labels that identify them in the screen layout. In this way, in most cases, you can also reference a field in the screen using its label (for example "UserID"). Keep in mind that no blank spaces or characters outside the A-Z range are allowed for label references. For example, if a field in the screen is labeled "NEXT COMMAND:", it will be referenced as "NEXTCOMMAND".

If an invalid index value or field name is specified, a *Null* value will be returned.

Related Topics

- [EditFields Collection](#)

5.1.1.1.6 HTML

The HTML property returns the current emulation in an HTML page for each screen.

Syntax

[*string* =] Display.[HTML](#) ([*top row*], [*left col*], [*bottom row*], [*right col*])

Remarks

The parameters mentioned above are optional; if they are not specified, the whole screen will be returned. Otherwise, you can specify the screen area you want to get, giving the corresponding screen coordinates.

5.1.1.1.7 ScreenId

The ScreenId property returns a sequential integer containing the screen number.

Syntax

[*Integer* =] Display.[ScreenId](#)

Remarks

When a connection is started, screens are subsequently numbered beginning at 0 (zero). This property is commonly used for error trapping purposes.

5.1.1.1.8 ScreenRow

The `ScreenRow` property returns the string contained in any given row in the current screen.

Syntax

```
[String =] Display.ScreenRow(Row As Integer)
```

Remarks

If an invalid *row* value is specified, an error will be returned.

Related Topics

- [Text Property](#)

5.1.1.1.9 Text

The `Text` property returns a string of a specific length contained in any given row and column combination in the current screen.

Syntax

```
[string =] Display.Text([row], [col], [len])
```

or

```
[string =] Display.Text(R[row]C[col]L[len])
```

Remarks

The *row* and *col* parameters specify the screen coordinates of the string. The *len* parameter specifies the length of the string.

Related Topics

- [ScreenRow Property](#)

5.1.1.2 Methods

For information about a particular method of the Display Object, go to any of the following links:

- [Connect](#)
- [Disconnect](#)
- [Press](#)
- [PressAndWait](#)
- [SendAid](#)
- [SendKeys](#)
- [Wait](#)
- [WaitFor](#)
- [WaitForConnect](#)
- [WaitForDisconnect](#)
- [WaitForScreen](#)
- [WaitForNewScreen](#)
- [WaitForUnlock](#)
- [WaitForNewUnlock](#)

5.1.1.2.1 Connect

The Connect method establishes a synchronous connection to the Telnet server.

Syntax

```
[boolean] = Display.Connect([timeout])
```

Remarks

The *timeout* parameter is optional and is measured in milliseconds. This method returns 'True' if the connection is successfully established and 'False' if the *timeout* period expires.

If you specify a value in the *timeout* parameter, the value of the WaitTimeout property will not have any effect.

Related Topics

- [Wait Method](#)
- [WaitForConnect Method](#)

5.1.1.2.2 Disconnect

The Disconnect method terminates the connection to the Telnet server.

Syntax

```
[boolean] = Display.Disconnect([timeout])
```

Remarks

The *timeout* parameter is optional and is measured in milliseconds. This method returns 'True' if the connection is successfully established and 'False' if the *timeout* period expires.

If you specify a value in the *timeout* parameter, the value of the WaitTimeout property will not have any effect.

Related Topics

- [Wait Method](#)
- [WaitForConnect Method](#)

5.1.1.2.3 Press

The Press method sends an Attention Identifier (AID) key to the terminal and continues code execution regardless of the system's current blocking state.

Syntax

```
Display.Press(AidKey As String)
```

Related Topics

- [PressAndWait Method](#)

5.1.1.2.4 PressAndWait

The PressAndWait method sends an Attention Identifier (AID) key to the terminal and blocks code execution until the system gets unlocked.

Syntax

```
Display.PressAndWait(AidKey As String)
```

Related Topics

- [Press Method](#)

5.1.1.2.5 SendAid

The SendAid method sets an Attention Identifier Key and sends the modified fields to the host. Modified fields can be input (unprotected) fields, or sometimes protected fields that have the 'Modified' property set to 'True'.

Syntax

Display.[SendAid](#)(*aidkey* As String)

Remarks

This method works like the AidKey property, but it takes a string with the name of the AID key as a parameter. You can specify only one key for each call to the method.

The following table lists the possible string values:

String Value	Meaning
Enter	Enter key
PF1	PF1 Key
PF2	PF2 Key
PF3	PF3 Key
PF4	PF4 Key
PF5	PF5 Key
PF6	PF5 Key
PF7	PF7 Key
PF8	PF8 Key
PF10	PF10 Key
PF11	PF11 Key
PF12	PF12 Key
PF13	PF13 Key
PF14	PF14 Key
PF15	PF15 Key
PF16	PF16 Key
PF17	PF17 Key
PF18	PF18 Key
PF19	PF19 Key
PF20	PF20 Key
PF21	PF21 Key
PF22	PF22 Key
PF23	PF23 Key
PF24	PF24 Key
PA1	PA1 Key
PA2	PA2 Key

PA3	PA3 Key
PA4	PA4 Key
PA5	PA5 Key

5.1.1.2.6 SendKeys

The SendKeys method can be used to send key sequences to the mainframe (including AID keys). By using special codes, you can send several kinds of keys. These codes consist of an escape character ("@"), and a mnemonic code that corresponds to the supported host functions.

Syntax

Display.[SendKeys](#)(keys As String)

Remarks

The following table lists the functions keys and its corresponding codes.

Value	Meaning
@A@Q	Attention
@<	Backspace
@B	BackTab (Left Tab)
@C	Clear
@E	Enter
@F	Erase Field
@A@Q	Sys Request
@T	Tab (Right Tab)
@x	PA1
@y	PA2
@z	PA3
@1	PF1
@2	PF2
@3	PF3
@4	PF4
@5	PF5
@6	PF6
@7	PF7
@8	PF8
@9	PF9
@a	PF10
@b	PF11
@c	PF12
@d	PF13
@e	PF14

@f	PF15
@g	PF16
@h	PF17
@i	PF18

This syntax complements the [SendKeys](#), adding the possibility to send several key sequences separated by aid keys and waits (@W). This is the main difference between both SendKeys methods. This means that you can navigate through several screens with a single method call.

Related Topics

- [SendAid Method](#)

5.1.1.2.7 Wait

This Wait method basically waits until the host enters the unlocked state and is ready to receive input data.

Syntax

```
[boolean] = Display.Wait([timeout])
```

Remarks

The *timeout* parameter is optional and is measured in milliseconds. This method returns 'True' if the wait is successful and 'False' if the *timeout* period expires.

This method uses alternatively the [WaitForConnect](#), [WaitForUnlock](#) and [WaitForNewScreen](#) methods, according to the current state of the host (disconnected, locked or unlocked).

If you specify a value in the *timeout* parameter, the value of the WaitTimeout property will not have any effect.

Related Topics

- [WaitForConnect Method](#)
- [WaitForUnlock Method](#)
- [WaitForNewScreen Method](#)

5.1.1.2.8 WaitFor

The WaitFor method waits for a screen that contains the specified string.

Syntax

```
[boolean] = Display.WaitFor(string, [timeout])
```

Remarks

The *timeout* parameter is optional and is measured in milliseconds. This method returns 'True' if the wait is successful and 'False' if the *timeout* period expires.

If you specify a value in the *timeout* parameter, the value of the WaitTimeout property will not have any effect.

Related Topics

- [WaitForScreen Method](#)
- [WaitForUnlock Method](#)
- [WaitForNewScreen Method](#)

5.1.1.2.9 WaitForConnect

The WaitForConnect method waits until the Telnet connection is successfully opened.

Syntax

```
[boolean] = Display.WaitForConnect([timeout])
```

Remarks

The *timeout* parameter is optional and is measured in milliseconds. This method returns 'True' if the wait is successful and 'False' if the *timeout* period expires.

If you specify a value in the *timeout* parameter, the value of the WaitTimeout property will not have any effect.

5.1.1.2.10 WaitForDisconnect

This method waits until the telnet connection is successfully closed, or the timeout period expires.

Syntax

```
[Boolean] = Display.WaitForDisconnect([TimeOut])
```

The *TimeOut* parameter is measured in milliseconds.

Remarks

The Timeout parameter is optional. This method returns True if the wait is successful and False if the Timeout period expires.

If you pass a timeout value in the Timeout parameter of this method, the value of the property WaitTimeout will not have any effect.

Related Topics

- [Wait](#)

5.1.1.2.11 WaitForScreen

The WaitForScreen method waits for the first screen displayed after the system returns from the locked state.

Syntax

```
[boolean] = Display.WaitForScreen([timeout])
```

Remarks

This method is similar to the [WaitForNewScreen](#) method, except that the screen to wait for can be any screen and not just a new one.

The *timeout* parameter is optional and is measured in milliseconds. This method returns 'True' if the wait is successful and 'False' if the *timeout* period expires.

If you specify a value in the *timeout* parameter, the value of the WaitTimeout property will not have any effect.

Related Topics

- [WaitForNewScreen](#)

5.1.1.2.12 WaitForNewScreen

The WaitForNewScreen method waits until a new screen is displayed.

Syntax

```
[boolean] = Display.WaitForNewScreen([timeout])
```

Remarks

This method is similar to the [WaitForScreen](#) method, except that in WaitForNewScreen, the screen to wait for has to be a new one.

The *timeout* parameter is optional and is measured in milliseconds. This method returns 'True' if the wait is successful and 'False' if the *timeout* period expires.

If you specify a value in the *timeout* parameter, the value of the WaitTimeout property will not have any effect.

Related Topics

- [WaitForScreen Method](#)

5.1.1.2.13 WaitForUnlock

The WaitForUnlock method waits until the host enters the unlocked state and is ready to receive input data.

Syntax

```
[boolean] = Display.WaitForUnlock([timeout])
```

Remarks

This method is similar to the [WaitForNewUnlock](#) method, except that the unlock to wait for can be any unlock and not just a new one.

The *timeout* parameter is optional and is measured in milliseconds. This method returns 'True' if the wait is successful and 'False' if the *timeout* period expires.

If you specify a value in the *timeout* parameter, the value of the WaitTimeout property will not have any effect.

Related Topics

- [WaitForNewUnlock](#)

5.1.1.2.14 WaitForNewUnlock

The WaitForNewUnlock method waits until a new system unlock arrives.

Syntax

```
[boolean] = Display.WaitForNewUnlock([timeout])
```

The *TimeOut* parameter is measured in milliseconds.

Remarks

This method is similar to the [WaitForUnlock](#) method, except that the unlock to wait for has to be a new one.

The *timeout* parameter is optional and is measured in milliseconds. This method returns 'True' if the wait is successful and 'False' if the *timeout* period expires.

If you specify a value in the *timeout* parameter, the value of the WaitTimeout property will not have any effect.

Related Topics

- [WaitForUnlock](#)

5.2 HLLAPI Reference

The HLLAPI functionality for TN3270 and TN5250 is provided through the file zhllap32.dll. This is where the entry points have to taken from. The zhllap32.dll is installed and registered in z/Scope's installation directory.

For a complete reference of common functions and WinHLLAPI extension functions, refer to the following topics:

- [EHLLAPI Functions](#)
- [WinHLLAPI Extension Functions](#)

5.2.1 EHLLAPI Functions

For information about a particular function, refer to any of the following links:

- [Cancel File Transfer \(92\)](#)
- [Change PS Window Name \(106\)](#)
- [Change Switch List LT Name \(105\)](#)
- [Connect Presentation Space \(1\)](#)
- [Connect Window Services \(101\)](#)
- [Convert Position or Convert RowCol \(99\)](#)
- [Copy Field to String \(34\)](#)
- [Copy OIA \(13\)](#)
- [Copy Presentation Space \(5\)](#)
- [Copy Presentation Space to String \(8\)](#)
- [Copy String to Field \(33\)](#)
- [Copy String to Presentation Space \(15\)](#)
- [Disconnect Presentation Space \(2\)](#)
- [Disconnect Window Service \(102\)](#)
- [Find Field Length \(32\)](#)
- [Find Field Position \(31\)](#)
- [Get Key \(51\)](#)
- [Lock Presentation Space API \(60\)](#)
- [Lock Window Services API \(61\)](#)
- [Pause \(18\)](#)
- [Post Intercept Status \(52\)](#)
- [Query Additional Field Attribute \(45\)](#)
- [Query Close Intercept \(42\)](#)
- [Query Communication Event \(81\)](#)
- [Query Cursor Location \(7\)](#)
- [Query Field Attribute \(14\)](#)
- [Query Host Update \(24\)](#)
- [Query Session Status \(22\)](#)
- [Query Sessions \(10\)](#)
- [Query System \(20\)](#)
- [Query Window Coordinates \(103\)](#)
- [Receive File \(91\)](#)
- [Release \(12\)](#)
- [Reserve \(11\)](#)
- [Reset System \(21\)](#)

- [Search Field \(30\)](#)
- [Search Presentation Space \(6\)](#)
- [Send File \(90\)](#)
- [Send Key \(3\)](#)
- [Set Cursor \(40\)](#)
- [Set Session Parameters \(9\)](#)
- [Start Close Intercept \(41\)](#)
- [Start Communication Notification \(80\)](#)
- [Start Host Notification \(23\)](#)
- [Start Keystroke Intercept \(50\)](#)
- [Start Playing Macro \(110\)](#)
- [Stop Close Intercept \(43\)](#)
- [Stop Communication Notification \(82\)](#)
- [Stop Host Notification \(25\)](#)
- [Stop Keystroke Intercept \(53\)](#)
- [Wait \(4\)](#)
- [Window Status \(104\)](#)

5.2.1.1 Cancel File Transfer (92)

The Cancel File Transfer function causes any current EHELLAPI initiated Send File or Receive File for the specified session to immediately return.

3270	5250
Yes	Yes

Prerequisite Calls

[Send File \(90\)](#) or [Receive File \(91\)](#)

Call Parameters

	Enhanced Interface
Function Number	Must be 92
Data String	1-character short name of the host presentation space. A blank or null indicates request for updates to the host-connected presentation space
Length	4 is implied
PS Position	NA

The calling data structure contains these elements

Byte	Definition
1	A 1-character presentation space short name (PSID)
2-4	Reserved

Return Parameters

Return Code	Definition
0	The function was successful

1	An incorrect PSID was specified
8	No prior call to Start Communication Notification (80) function was called for the PSID
9	A systemerror was encountered

Remarks

Since both [Send File \(90\)](#) and [Receive File \(91\)](#) are blocking calls, this function must always be issued on a different thread.

5.2.1.2 Change PS Window Name (106)

The Change PS Window Name function allows the application to specify a new name for the presentation space window or reset the presentation space window to the default name.

3270	5250
Yes	Yes

Prerequisite Calls

[Connect Window Services \(101\)](#)

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 106	
Data String	See the following table	
Length	Must be specified (See note.)	Must be 68
PS Position	NA	

Note: The data string length must be specified (normally 3-63 for PC/3270, 4-63 for PC400, 68 for enhanced interface).

The calling data string can contain:

Byte		Definition
	Standard	Enhanced
1	1	A 1-character presentation space short name (PSID)
		2-4
		Reserved
2	5	A change request option value, select one of: <ul style="list-style-type: none"> • X'01' for changing the presentation space window name. • X'02' for resetting the presentation space window name.
3-63	6-66	An ASCII string of from 1 (for PC/3270) or 2 (for PC400) to 61 bytes including a terminator byte. The ASCII string must end with a NULL character. This string must contain at least one non-NULL

		character followed by a NULL character.
	67-68	Reserved

Return Parameters

Return Code	Explanation
0	The Change PS Window Name function was successful.
1	An incorrect host presentation space short session ID was specified, or the host presentation space was not connected.
2	An error was made in specifying parameters.
9	A system error occurred.
12	The session stopped.

Remarks

A string is ended at the first NULL character found. The NULL character overrides the specified string length. If the NULL character is not at the end of the specified length, the last byte at the specified length is replaced by a NULL character, and the remainder of the data string is lost. If the NULL character is found before the specified length, the string is truncated at that point, and the remainder of the data string is lost.

If the application fails to reset the presentation space name before exiting, the exit list processing resets the name.

5.2.1.3 Change Switch List LT Name (105)

The Change Switch List LT Name function allows the application to change or reset a switch list for a selected logical terminal (LT). The application must specify on the call the name to be inserted in the switch list.

Note: This is for compatibility with Communication Manager EHLLAPI, and has the same result as the [Change PS Window Name \(106\)](#) function.

3270	5250
Yes	Yes

Prerequisite Calls

[Connect Window Services \(101\)](#)

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 105	
Data String	See the following table	
Length	Normally 4-63	Must be 68
PS Position	NA	

The calling data string can contain:

Byte		Definition
Standard	Enhanced	
1	1	A 1-character presentation space short name (PSID)
	2-4	Reserved
2	5	A change request option; select: <ul style="list-style-type: none"> • X'01' for changing a switch list LT name • X'02' for resetting a switch list LT name
3-63	6-66	An ASCII string of 2 to 61 bytes including a terminator byte. The ASCII string must end with a NULL character. This string must contain at least one non-NULL character followed by a NULL character.
	67-68	Reserved

Return Parameters

Return Code	Explanation
0	The Change Switch List LT Name function was successful.
1	An incorrect host presentation space short session ID was specified, or the host presentation space was not connected.
2	An error was made in specifying parameters.
9	A system error occurred.
12	The session stopped.

Remarks

A string is ended at the first NULL character found. The NULL character overrides the specified string length. If the NULL character is not at the end of the specified length, the last byte at the specified length is replaced by a NULL character, and the remainder of the data string is lost. If the NULL character is found before the specified length, the string is truncated at that point, and the remainder of the data string is lost.

If the application fails to reset the switch list LT name before exiting, the exit list processing resets the name.

5.2.1.4 Connect Presentation Space (1)

The Connect Presentation Space function establishes a connection between your EHLAPI application program and the host presentation space.

3270	5250
Yes	Yes

Prerequisite Calls

There are no prerequisite calls for this function.

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 1	
Data String	1-character short name of the host presentation space	
Length	1 is implied	Must be 4
PS Position	NA	

The calling data string can contain:

Byte		Definition
Standard	Enhanced	
1	1	A 1-character presentation space short name (PSID)
	2-4	Reserved

Return Parameters

The Connect Presentation Space function sets the return code to indicate the status of the attempt and, if successful, the status of the host presentation space.

Return Code	Explanation
0	The Connect Presentation Space function was successful; the host presentation space is unlocked and ready for input.
1	An incorrect host presentation space ID was specified. The specified session either does not exist or is a logical printer session. This return code could also mean that the API Setting for DDE/EHLLAPI is not set on.
4	Successful connection was achieved, but the host presentation space is busy.
5	Successful connection was achieved, but the host presentation space is locked (input inhibited).
9	A system error was encountered.
11	This resource is unavailable. The host presentation space is already being used by another system function.

Remarks

1. The Connect Presentation Space function is affected by the CONLOG/CONPHYS session option.
2. An EHLLAPI application cannot be connected to multiple presentation spaces concurrently. Calls requiring the Connect Presentation Space function as a prerequisite use the currently connected presentation space. For example, if an application is connected to presentation space A, B, and C in that order, the application must connect to B or A again to issue functions.
3. Each thread that requests a Connect Presentation Space must have a corresponding [Disconnect Presentation Space \(2\)](#), or one of the threads must issue a [Reset System \(21\)](#), which affects all threads and disconnects any remaining connections.

4. More than one EHLLAPI application can share a presentation space, if the applications support sharing (that is, if they were developed to work together and if they exhibit predictable behavior) and have compatible read/write access and keyword options as set in the [Set Sessions Parameters \(9\)](#) function. For more information, see [Set Session Parameters \(9\)](#).
5. Because the Connect Presentation Space and [Start Keystroke Intercept \(50\)](#) functions share common subsystem functions, successful requests by an application to share either of these functions for the same session can affect the request of these two functions by other applications. For example, if application A successfully requests a Connect Presentation Space for a session with Write_Read access and KEY\$abcdefgh as the keyword, a request by application B to Connect Presentation Space for a session and Start Keystroke Intercept is successful only if both applications have set compatible read/write options.

5.2.1.5 Connect Window Services (101)

The Connect Window Services function allows the application to manage the presentation space windows. Only one EHLLAPI application at a time can be connected to a presentation space for window services.

An EHLLAPI application can connect to more than one presentation space concurrently for window services.

3270	5250
Yes	Yes

Prerequisite Calls

There are no prerequisite calls for this function.

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 101	
Data String	1-character short session ID of the host presentation space	
Length	1 is implied	Must be 4
PS Position	NA	

The calling data string can contain:

Byte		Definition
Standard	Enhanced	
1	1	A 1-character presentation space short name (PSID)
	2-4	Reserved

Return Parameters

Return Code	Explanation
0	The Connect Window Services function was successful.
1	An incorrect host presentation space short session ID was specified, or the Sessions Window Services manager was not connected. This return code could also mean that the API Setting for DDE/EHLLAPI is not set on.
9	A system error occurred.
10	The function is not supported by the emulation program.
11	This resource is unavailable. The host presentation space is already being used by another system function.

Remarks

1. An EHLLAPI application can be connected to multiple presentation space windows at the same time. The application can go back and forth between the connected presentation space windows without having to disconnect. For example, if an application is connected to presentation space windows A, B, and C, the application can access all of A, B, and C at the same time, and the other applications cannot access A, B, or C.
2. A Connect Window Services function is sufficient for the process. However, each thread that requests a Connect Window Services must have a corresponding [Disconnect Window Services \(102\)](#), or one of the threads must issue a [Reset System \(21\)](#), which affects all threads and disconnects any remaining connections.

5.2.1.6 Convert Position or Convert RowCol (99)

The Convert Position or Convert RowCol function converts the host presentation space positional value into the display row and column coordinates or converts the display row and column coordinates into the host presentation space positional value. This function does not change the cursor position.

3270	5250
Yes	Yes

Prerequisite Calls

There are no prerequisite calls for this function.

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 99	
Data String	Host presentation space short name and P for the Convert Position function (for example, AP converts the presentation space position of session A); or Host presentation space short name and R for the Convert RowCol function (for example, AR converts the row and column coordinates of session A).	
Length	Row, when R is specified as the second character in the data string parameter. The lower limit for valid input is 1. The upper limit for valid input depends on how your host presentation space is configured. See 'Remarks'. NA when P is specified as the second character in the data string parameter.	
PS Position	Column, when R is specified as the second character in the data string parameter. The lower limit for valid input is 1. The upper limit for valid input	

	ranges from 24 to 43 depending on how your host presentation space is configured. See Remarks. Host presentation space position, when P is specified as the second character in the data string parameter. The lower limit for valid input is 1. The upper limit for valid input ranges from 1920 to 3564 depending on how your host presentation space is configured. See 'Remarks'.
--	---

The calling data string can contain:

Byte		Definition
Standard	Enhanced	
1	1	A 1-character presentation space short name (PSID)
	2-4	Reserved
2	5	Convert option P or R
	6-8	Reserved

Return Parameters

This function returns a length and a return code.

Length

For the Convert Position function (P as the second character in the calling data string), a number between 1 and 43 (for PC/3270) or 27 (for PC400) is returned. This value is the number of the row that contains the PS position contained in the calling PS position parameter. The upper limit can be smaller than 43 (for PC/3270) or 27 (for PC400) depending on how the host presentation space is configured.

For the Convert RowCol function (R as the second character in the calling data string), a value of 0 indicates an error in the input value for row (calling length parameter).

Return Code

The Convert Position or RowCol function is the exception to the rule that the fourth return parameter always contains a return code. For this function, the value returned in the fourth parameter is called a status code. This status code can contain data or a return code. Your application must provide for processing of this status code to prevent unpredictable results or an error.

- If the value of the fourth parameter is 0, 9998, or 9999, it is a return code.
- For the Convert Position function (P as the second character of the calling data string), a value in the range of 1-132 is the number of the column that contains the PS position passed in the calling PS Position parameter. The upper limit can be smaller than 132 depending on how the host presentation space is configured.
- For the Convert RowCol function (R as the second character of the calling data string), a value in the range of 1-3564 represents the host presentation space position that corresponds to the row and column values passed in the calling length and PS position parameters, respectively. The upper limit can be smaller than 3564 depending on how the host presentation space is configured.

The following status codes are defined:

Status Code	Explanation
0	This is an incorrect PS position or column.
>0	This is the PS position or column.
9998	An incorrect host presentation space ID was specified or a system error occurred.
9999	Character 2 in the data string is not P or R.

Remarks

To find out how many rows and columns are in your presentation space, examine the returned data string parameter for the [Query Session Status \(22\)](#) function.

5.2.1.7 Copy Field to String (34)

The Copy Field to String function transfers characters from a field in the host-connected presentation space into a string.

The Copy Field to String function translates the characters in the host source presentation space into American National Standard Code for Information Interchange (ASCII). Attribute bytes and other characters not represented in ASCII normally are translated into blanks.

3270	5250
Yes	Yes

Prerequisite Calls

[Connect Presentation Space \(1\)](#)

Call Parameters

	Standard Interface
Function Number	Must be 34
Data String	Preallocated target data string. When the Set Session Parameters (9) function with Extended Attribute Bytes (EAB) option is issued, the length of the data string must be at least twice the length of the field. DBCS Only: When Extended Attributes Double-byte (EAD) option is specified, the length of the data string must be at least three times the length of the field. When both EAB and EAD options are specified, the length of the data string must be at least four times the length of the field.
Length	Number of bytes to copy (the length of the data string).
PS Position	Identifies the target field. This can be the PS position of any byte within the target field. Copy always starts at the beginning of the field.

Return Parameters

This function returns a data string, length, and a return code.

Data String

A string containing data from the identified field in the host presentation space. The

first byte in the returned data string is the beginning byte of the identified field in the host presentation space. The number of bytes in the returned data string is determined by the smaller of:

- Number of bytes specified in the calling length parameter
- Number of bytes in the identified field in the host presentation space

Length

The length of the data returned.

Return Code

The following codes are defined:

Return Code	Explanation
0	The Copy Field to String function was successful.
1	Your program is not connected to a host session.
2	An error was made in specifying parameters.
6	The data to be copied and the target field are not the same size. The data is truncated if the string length is smaller than the field copied.
7	The host presentation space position is not valid.
9	A system error was encountered.
24	Unformatted host presentation space.

Remarks

1. The field position and length information can be found by using the [Find Field Position \(31\)](#) and [Find Field Length \(32\)](#) functions. The Copy Field to String function can be used with either protected or unprotected fields, but only in a field-formatted host presentation space.
2. The copy is ended when one of the following conditions is encountered when the end of the field is reached and when the length of the target string is exceeded.
3. DBCS Only: If the target string is ended at the higher byte of the DBCS character, the byte is translated into a blank. If the EAD option is set to on, three bytes are returned for each character. If both the EAB and EAD options are set to on, four bytes are returned for each character. When the field wraps at the end of the presentation space, wrapping occurs when the end of the presentation space is reached.
4. DBCS Only: The [Set Session Parameters \(9\)](#) function EAD option is used with this function to return a 2-byte EAD. If the EAD option is specified instead of the EAB option, EAD is returned preceding each character. If both the EAB and EAD options are specified, EAD is returned preceding the EAB.
5. An EAB can be returned when the [Set Session Parameters \(9\)](#) function EAB option is used. EAB is related to each character in the presentation space and is returned preceding each character.
6. The Copy Field to String function is affected by the ATTRB/NOATTRB/NULLATTRB, the EAB/NOEAB, the XLATE/NOXLATE, the DISPLAY/NODISPLAY, the DISPLAY/NODISPLAY, the EAD/NOEAD (for DBCS only), and the NOSO/SPACESO/SO (for DBCS only) session options.
7. To use this function, preallocate memory to receive the returned data string parameter. The statements required to preallocate this memory vary depending on the language in which your application is written.

5.2.1.8 Copy OIA (13)

The Copy OIA function returns the current operator information area (OIA) data from the host-connected presentation space.

The OIA is located under the bottom dividing line of the screen and is used to display session status information about the connection between the workstation and the host.

3270	5250
Yes	Yes

Prerequisite Calls

[Connect Presentation Space \(1\)](#)

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 13	
Data String	Preallocated target data string	
Length	103	104
PS Position	NA	

Return Parameters

This function returns a data string and a return code.

Data String

A 103-byte string for 16-bit and 104-byte string for 32-bit.

Return Code

The following codes are defined:

Return Code	Explanation
0	OIA data is returned. The target presentation space is unlocked.
1	Your program is not connected to a host session.
2	An error was made in specifying string length. OIA data was not returned.
4	OIA data is returned. The target presentation space is busy.
5	OIA data is returned. The target presentation space is locked. (Input inhibited)
9	An internal system error was encountered. OIA data was not returned.

Remarks

1. The OIA Group consists of the bits that show the status of the connected sessions. The group is categorized by the represented host function. (For

example, Group 8 consists of the bits that show all conditions of the input inhibit in the session.) The states of each group are ordered so that the high-order bits represent the indicators of higher priority. That is, bit 7 has priority over bit 0. Therefore, if more than one state is active within a group, the state with the highest priority is the active state within that group.

- To use this function, preallocate memory to receive the returned data string parameter. The statements required to preallocate this memory vary depending on the language in which your application is written.

5.2.1.9 Copy Presentation Space (5)

The Copy Presentation Space function copies the contents of the host-connected presentation space into a data string that you define in your EHLAPI application program.

The Copy Presentation Space function translates the characters in the host source presentation space into ASCII. Attribute bytes and other characters not represented in ASCII normally are translated into blanks. If you do not want the attribute bytes translated into blanks, you can override this translation with the ATTRB option under the [Set Session Parameters \(9\)](#) function.

3270	5250
Yes	Yes

Prerequisite Calls

[Connect Presentation Space \(1\)](#)

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 5	
Data String	Preallocated target string the size of your host presentation space. This can vary depending on how your host presentation space is configured. When the Set Session Parameters (9) function with the EAB option is issued, the length of the data string must be at least twice the length of the presentation space. DBCS Only: When the EAD option is specified, the length of the data string must be at least three times the length of the presentation space. When both the EAB and EAD options are specified, the length of the data string must be at least four times the length of the presentation space.	
Length	NA (the length of the host presentation space is implied).	
PS Position	NA.	

Return Parameters

This function returns a data string, length, and a return code.

Data String

Contents of the connected host presentation space.

Length

Length of the data copied.

Return Code

The following codes are defined:

Return Code	Explanation
0	The host presentation space contents were copied to the application program. The target presentation space was active, and the keyboard was unlocked.
1	Your program is not connected to a host session.
4	The host presentation space contents were copied. The connected host presentation space was waiting for host response.
5	The host presentation space was copied. The keyboard was locked.
9	A system error was encountered.

Remarks

1. An EAB can be returned when the Set Session Parameters (9) function EAB option is used. EAB is related to each character in the presentation space and is returned preceding each character.
2. DBCS Only: The [Set Session Parameters \(9\)](#) function EAD option is used with this function to return a 2-byte EAD. If the EAD option is specified instead of the EAB option, EAD is returned preceding each character. If both the EAB and EAD options are specified, EAD is returned preceding the EAB. If the start position of the copy is at the second byte in the double-byte character, or the end position is at the first byte in the double-byte character, the bytes are translated into blanks.
4. The Copy Presentation Space function is affected by the following session options:
 - ATTRB/NOATTRB/NULLATTRB
 - EAB/NOEAB
 - XLATE/NOXLATE
 - BLANK/NOBLANK
 - DISPLAY/NODISPLAY
 - EAD/NOEAD (for DBCS only)
 - NOSO/SPACESO/SO (for DBCS only)
 - EXTEND_PS/NOEXTEND_PS
5. If you want to copy only a portion of the host presentation space, use the [Copy Presentation Space to String \(8\)](#) function.
6. To use this function, preallocate memory to receive the returned data string parameter. The statements required to preallocate this memory vary depending on the language in which your application is written.

5.2.1.10 Copy Presentation Space to String (8)

The Copy Presentation Space to String function is used to copy all or part of the host-connected presentation space into a data string that you define in your EHLAPI application program.

The input PS position is the offset into the host presentation space. This offset is

based on a layout in which the upper-left corner (row 1/column 1) is location 1 and the bottom-right corner is 3564, which is the maximum screen size for the host presentation space. The value of PS Position + (Length - 1) cannot exceed the configured size of your host presentation space.

The Copy Presentation Space to String function translates the characters in the host source presentation space into ASCII. Attribute bytes and other characters not represented in ASCII normally are translated into blanks. If you do not want the attribute bytes translated into blanks, you can override this translation with the ATTRB option under the [Set Session Parameters \(9\)](#) function.

3270	5250
Yes	Yes

Prerequisite Calls

[Connect Presentation Space \(1\)](#)

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 8	
Data String	Preallocated target string the size of your host presentation space. When the Set Session Parameters (9) function with the EAB option is issued, the length of the data string must be at least twice the length of the presentation space. DBCS Only: When the EAD option is specified, the length of the data string must be at least three times the length of the presentation space. When both the EAB and EAD options are specified, the length of the data string must be at least four times the length of the presentation space.	
Length	Length of the target data string.	
PS Position	Position within the host presentation space of the first byte in your target data string.	

Return Parameters

This function returns a data string and a return code.

Data String

Contents of the host presentation space.

Return Code

The following codes are defined:

Return Code	Explanation
0	The host presentation space contents were copied to the application program. The target presentation space was active, and the keyboard was unlocked.
1	Your program is not connected to a host session.
2	An error was made in specifying string length, or the sum of (Length - 1) + PS position is greater

	than the size of the connected host presentation space.
4	The host presentation space contents were copied. The host presentation space was waiting for host response.
5	The host presentation space was copied. The keyboard was locked.
7	The host presentation space position is not valid.
9	A system error was encountered.

Remarks

1. An EAB can be returned when the [Set Session Parameters \(9\)](#) function EAB option is used. EAB is related to each character in the presentation space and is returned following each character.
2. DBCS Only: The [Set Session Parameters \(9\)](#) function EAD option is used with this function to return a 2-byte EAD. If the EAD option is specified instead of the EAB option, EAD is returned preceding each character. If both the EAB and EAD options are specified, EAD is returned following the EAB.
3. If the start position of the copy is at the second byte in the double-byte character, or the end position is at the first byte in the double-byte character, the bytes are translated into blanks. If the EAD option is set to on, three bytes are returned for each character. If both the EAB and EAD options are set to on, four bytes are returned for each character.
4. The Copy Presentation Space to String function is affected by the following options:
 - ATTRB/NOATTRB/NULLATTRB
 - EAB/NOEAB
 - XLATE/NOXLATE
 - BLANK/NOBLANK
 - DISPLAY/NODISPLAY
 - EAD/NOEAD (for DBCS only)
 - NOSO/SPACESO/SO (for DBCS only)
 - EXTEND_PS/NOEXTEND_PS
5. If the target data string provided is not large enough to hold the requested number of bytes, the copy ends successfully (RC=0, 4, or 5) when the end of the target data string is reached.
6. To use this function, preallocate memory to receive the returned data string parameter. The statements required to preallocate this memory vary depending on the language in which your application is written.

5.2.1.11 Copy String to Field (33)

The Copy String to Field function transfers a string of characters into a specified field in the host-connected presentation space. This function can be used only in a field-formatted host presentation space.

3270	5250
Yes	Yes

Prerequisite Calls

[Connect Presentation Space \(1\)](#)

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 33	
Data String	String containing the data to be transferred to a target field in the host presentation space.	
Length	Length, in number of bytes, of the source data string. Overridden if in EOT mode.	
PS Position	Identifies the target field. This can be the PS position of any byte within the target field. Copy always starts at the beginning of the field.	

Return Parameters

Return Code	Explanation
0	The Copy String to Field function was successful.
1	Your program is not connected to a host session.
2	Parameter error or zero length for copy.
5	The target field was protected or inhibited, or incorrect data was sent to the target field (such as a field attribute).
6	Copy was completed, but data is truncated.
7	The host presentation space position is not valid.
9	A system error was encountered.
24	Unformatted host presentation space.

Remarks

- The Copy String to Field function is affected by the following options:
 - STRLEN/STREOT
 - EOT
 - EAB/NOEAB
 - XLATE/NOXLATE
 - PUTEAB/NOPUTEAB
- The string to be transferred is specified with the calling data string parameter. The string ends when one of these three conditions is encountered:
 - When an end-of-text (EOT) delimiter is encountered in the string if EOT mode was selected using the [Set Session Parameters \(9\)](#) function.
 - When the number specified in the length is reached if not in EOT mode.
 - When an end-of-field is encountered in the field.

Note: If the field at the end of the host presentation space wraps, wrapping occurs when the end of the presentation space is reached.
- The keyboard mnemonics (see [Send Key \(3\)](#) function) cannot be sent using the Copy String to Field function.
- The first byte of the data to be transferred is always placed at the beginning of the field that contains the specified PS position.
- DBCS Only: Double-byte characters can be included as a part of the string.

5.2.1.12 Copy String to Presentation Space (15)

The Copy String to Presentation Space function copies an ASCII data string directly into the host presentation space at the location specified by the PS position calling parameter.

3270	5250
Yes	Yes

Prerequisite Calls

[Connect Presentation Space \(1\)](#)

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 15.	
Data String	String of ASCII data to be copied into the host presentation space.	
Length	Length, in number of bytes, of the source data string. Overridden if in EOT mode.	
PS Position	Position in the host presentation space to begin the copy, a value between 1 and the configured size of your host presentation space.	

Return Parameters

Return Code	Explanation
0	The Copy String to Presentation Space function was successful.
1	Your program is not connected to a host session.
2	Parameter error or zero length for copy.
5	The target presentation space is protected or inhibited, or incorrect data was sent to the target presentation space (such as a field attribute byte).
6	The copy was completed, but the data was truncated.
7	The host presentation space position is not valid.
9	A system error was encountered.

Remarks

- The Copy String to Presentation Space function is affected by the following options:
 - STRLEN/STREOT
 - EOT
 - EAB/NOEAB
 - XLATE/NOXLATE
 - PUTEAB/NOPUTEAB
 - EAD/NOEAD (for DBCS only)
 - NOSO/SPACESO/SO (for DBCS only)
 - EXTEND_PS/NOEXTEND_PS
- The keyboard mnemonics (see [Send Key \(3\)](#) function) cannot be sent using the Copy String to Presentation Space function.
- The string ends when an end-of-text (EOT) delimiter is encountered in the string if EOT mode was selected using the [Set Session Parameters \(9\)](#) function.
- Although the Send Key (3) function accomplishes the same purpose, this function responds with the prompt and enters a command more quickly. Because the Send Key (3) function emulates the terminal operator typing the data from the

keyboard, its process speed is slow for an application operating with a lot of data. This function provides a faster input path to the host.

5. The original data (the copied string) cannot exceed the size of the presentation space.
6. DBCS Only: Double-byte characters can be included as a part of the string.

5.2.1.13 Disconnect Presentation Space (2)

The Disconnect Presentation Space function drops the connection between your EHLLAPI application program and the host presentation space. Also, if a host presentation space is reserved using the [Reserve \(11\)](#) function, it is released upon execution of the [Disconnect Presentation Space](#) function.

3270	5250
Yes	Yes

Prerequisite Calls

[Connect Presentation Space \(1\)](#)

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 2	
Data String	NA	
Length	NA	
PS Position	NA	

Return Parameters

Return Code	Explanation
0	The Disconnect Presentation Space function was successful.
1	Your program was not currently connected to the host presentation space.
9	A system error was encountered.

Remarks

1. After the Disconnect Presentation Space function is called, functions that interact with the host-connected presentation space are no longer valid (for example, the [Send Key \(3\)](#), [Wait \(4\)](#), [Reserve \(11\)](#) and [Release \(12\)](#) functions).
2. Your EHLLAPI application should disconnect from the host presentation space before exiting.
3. The Disconnect Presentation Space function does not reset the session parameters to the defaults. Your EHLLAPI application must call the [Reset System \(21\)](#) function to accomplish this.

5.2.1.14 Disconnect Window Service (102)

The Disconnect Window Service function disconnects the window services connection between the EHELLAPI program and the specified host presentation space window.

3270	5250
Yes	Yes

Prerequisite Calls

[Connect Window Services \(101\)](#)

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 102	
Data String	See the following table	
Length	1	4
PS Position	NA	

Data String Contents

Byte		Definition
Standard	Enhanced	
1	1	A 1-character presentation space short name (PSID)
	2-4	Reserved

Return Parameters

Return Code	Explanation
0	The Disconnect Window Service function was successful.
1	Your program is not connected for Window Services.
9	A system error occurred.

Remarks

1. After the Disconnect Window Service function has been called, your application no longer manages the presentation space window.
2. Before exiting the application, you should request a Disconnect Window Service function for all presentation spaces that have been connected for Presentation Manager^(R) services. If the application exits with an outstanding connection for window services, the subsystem cancels the outstanding connection.

5.2.1.15 Find Field Length (32)

The Find Field Length function returns the length of a target field in the connected presentation space. This function can be used to find either protected or unprotected fields, but only in a field-formatted host presentation space.

This function returns the number of characters contained in the field identified using the call PS position parameter. This includes all characters from the beginning of the target field up to the character preceding the next attribute byte.

3270	5250
Yes	Yes

Prerequisite Calls

[Connect Presentation Space \(1\)](#)

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 32	
Data String	See the following table	
Length	NA	NA
PS Position	See note	

Note:

PS Position: Identifies the field within the host presentation space at which to start the Find. It can be the PS position of any byte within the field in which you desire the Find to start.

The calling 2-character data string can contain:

Code	Explanation
[RB][RB] or T [RB]	This field
P [RB]	The previous field, either protected or unprotected.
N [RB]	The next field, either protected or unprotected
NP	The next protected field
NU	The next unprotected field
PP	The previous protected field
PU	The previous unprotected field

Note:

[RB] represents a required blank.

Return Parameters

This function returns a length and a return code.

Length

The following lengths are valid:

Length	Explanation
= 0	When return code = 28, field length is 0. When return code = 24, host presentation space is not field formatted.
> 0	Required field length in the host presentation space.

Return Code

The following codes are defined:

Return Code	Explanation
0	The Find Field Length function was successful.
1	Your program is not connected to a host session.
2	A parameter error was encountered.
7	The host presentation space position is not valid.
9	A system error was encountered.
24	No such field was found.
28	Field length of 0 bytes.

Remarks

Except when [RB][RB] or T [RB] is used as the calling data string, if the field found is the same as the field from which the Find started, a return code of 24 is returned.

5.2.1.16 Find Field Position (31)

The Find Field Position function returns the beginning position of a target field in the host-connected presentation space. This function can be used to find either protected or unprotected fields but only in a field-formatted host presentation space.

3270	5250
Yes	Yes

Prerequisite Calls

[Connect Presentation Space \(1\)](#)

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 31	
Data String	See the following table	
Length	NA	NA
PS Position	See note	

Note:

PS Position: Identifies the field within the host presentation space at which to start

the Find. It can be the PS position of any byte within the field in which you want the Find to start.

The calling 2-character data string can contain:

Code	Explanation
[RB][RB] or T [RB]	This field
P [RB]	The previous field, either protected or unprotected
N [RB]	The next field, either protected or unprotected
NP	The next protected field
NU	The next unprotected field
PP	The previous protected field
PU	The previous unprotected field

Note:

[RB] represents a required blank.

Return Parameters

This function returns a length and a return code.

Length

The following lengths are valid:

Length	Explanation
= 0	When return code = 28, field length is 0. When return code = 24, host presentation space is not field-formatted.
> 0	Relative position of the requested field from the origin of the host presentation space. This position is defined to be the first position after the attribute byte.

Return Code

The following codes are defined:

Return Code	Explanation
0	The Find Field Position function was successful.
1	Your program is not connected to a host session.
2	A parameter error was encountered.
7	The host presentation space position is not valid.
9	A system error was encountered.
24	No such field was found.
28	Field length of 0 bytes.

Remarks

Except when [RB][RB] or T [RB] is used as the calling data string, if the field found is the same as the field from which the Find started, a return code of 24 is returned.

5.2.1.17 Get Key (51)

The Get Key function lets your EHLAPI application program retrieve a keystroke from a session specified by the [Start Keystroke Intercept \(50\)](#) function and either process, accept, or reject that keystroke. By placing this function in a loop, you can use it to intercept a string.

3270	5250
Yes	Yes

Prerequisite Calls

[Start Keystroke Intercept \(50\)](#)

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 51	
Data String	See the following table	
Length	8	12
PS Position	NA	

Data String Contents

Byte		Definition
Standard	Enhanced	
1	1	One of the following values: <ul style="list-style-type: none"> A 1-character presentation space short name (PSID) A blank or null indicating a function call for the host-connected presentation
	2-4	Reserved
2-8	5-11	Blanks that hold space for the symbolic representation of the requested data
	12	Reserved

Return Parameters

This function returns a data string and a return code.

Data String

See the following table:

Byte		Definition
Standard	Enhanced	
1	1	One of the following values: <ul style="list-style-type: none"> A 1-character presentation space short name (PSID) A blank or null indicating a function call for the host-connected presentation
	2-4	Reserved

2	5	An option code character, one of the following characters: <ul style="list-style-type: none"> • A for ASCII returned • M for keystroke mnemonic • S for special mnemonic
3-8	6-11	These 6 bytes of the preallocated buffer space are used internally to enqueue and dequeue keystrokes. Possible combinations include: <ul style="list-style-type: none"> • Byte 3 contains an ASCII character and byte 4 contains 'X'00' • Bytes 3 and 4 contain a double-byte character • Byte 3 contains the escape character (either @ or another character specified using the ESC=c option of function 9) and byte 4 contains a 1-byte abbreviation for a function. • Bytes 5 through 8 might be similar to bytes 3 and 4 if the returned ASCII mnemonic is longer than 2 bytes (for example, if the ASCII mnemonic represents Attn @A@Q, byte 5 contains @ and byte 6 contains Q). If not used, bytes 5 through 8 are set to zero (X'00').

Note:

The @ symbol is the default escape character. The value of the escape character can be set to any keystroke represented in ASCII by using the ESC=c option of the [Set Session Parameters \(9\)](#) function. If the escape character has been changed to another character using this option, the @ symbol in the following examples is replaced by the other character.

Return Code

The following codes are valid:

Return Code	Explanation
0	The Get Key function was successful.
1	An incorrect presentation space was specified.
5	You specified the AID only option under the Start Keystroke Intercept (50) function, and non-AID keys are inhibited by this session type when EHLLAPI tries to write incorrect keys to the presentation space.
8	No prior Start Keystroke Intercept (50) function was called for this presentation space.
9	A system error was encountered.
20	An undefined key combination was typed.
25	The requested keystrokes are not available on the input queue.
31	Keystroke queue overflowed and keystrokes were lost.

Remarks

- If a return code of 31 occurs for the Get Key function, either:
 - Increase the value of the calling length parameter for the [Start Keystroke Intercept \(50\)](#) function, or
 - Execute the Get Key function more frequently.

An intercepted keystroke occupies 3 bytes in the buffer. The next intercepted keystroke is placed in the adjacent three bytes. When the Get Key function retrieves a keystroke (first in first out, FIFO), the three bytes that it occupied are made available for another keystroke. By increasing the size of the buffer or the rate at which keystrokes are retrieved from the buffer, you can eliminate buffer overflow.

For the PC/3270, another way to eliminate return code 31 is to operate the PC/3270 emulator in the resume mode.
- You can use the [Send Key \(3\)](#) function to pass both original keystrokes and any others that your EHLLAPI application might need to the host-connected

- presentation space.
3. Keystrokes arrive asynchronously and are enqueued in the keystroke queue that you have provided in your EHELLAPI application program using the [Start Keystroke Intercept \(50\)](#) function.
 4. The Get Key function behaves like a read. When keystrokes are available, they are read into the data area that you have provided in your application.
 5. In the case of field support for a session, the application might be interested only in AID keys, for example the Enter key. If so, the [Start Keystroke Intercept \(50\)](#) function option code should be set to D (meaning for AID Keys only).
 6. To use this function, preallocate memory to receive the returned data string parameter. The statements required to preallocate this memory vary depending on the language in which your application is written.

5.2.1.18 Lock Presentation Space API (60)

The Lock Presentation Space API function allows the application to obtain or release exclusive control of the presentation space window over other Windows 32-bit applications. While locked, no other application can connect to the presentation space window.

Successful processing of this function with the Lock causes EHELLAPI presentation space window functions requested from other EHELLAPI applications to be queued until the requesting application unlocks the presentation space. Requests from the locking application are processed normally.

3270	5250
Yes	No

Prerequisite Calls

[Connect to Presentation Space \(1\)](#)

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 60	
Data String	See the following table	
Length	Must be 3	Must be 8
PS Position	NA	

Data String Contents

Byte		Definition
Standard	Enhanced	
1	1	A 1-character presentation space short name (PSID).
	2-4	Reserved.
2	5	One of the following characters: <ul style="list-style-type: none"> • L to lock the API. • U to unlock the API.

3	6	One of the following characters: <ul style="list-style-type: none"> • R to return if the presentation space is already locked by an application. • Q to queue the Lock request if the presentation space is already locked by an application.
	7-8	Reserved.

Return Parameters

Return Code	Explanation
0	The Lock Presentation Space API function was successful.
1	An incorrect host presentation space short session ID was specified or was not connected.
2	An error was made in specifying parameters.
9	A system error was encountered.
43	The API was already locked by another EHELLAPI application (on LOCK) or API not locked (on UNLOCK).

Remarks

The following EHELLAPI functions are queued when a lock is in effect:

- [Send Key \(3\)](#)
- [Copy Presentation Space \(5\)](#)
- [Search Presentation Space \(6\)](#)
- [Copy Presentation Space to String \(8\)](#)
- [Release \(11\)](#)
- [Reserve \(12\)](#)
- [Query Field Attribute \(14\)](#)
- [Copy String to Presentation Space \(15\)](#)
- [Search Field \(30\)](#)
- [Find Field Position \(31\)](#)
- [Find Field Length \(32\)](#)
- [Copy String to Field \(33\)](#)
- [Copy Field to String \(34\)](#)
- [Set Cursor \(40\)](#)
- [Send File \(90\)](#)
- [Receive File \(91\)](#)
- [Connect to Presentation Space \(1\)](#) with the CONPHYS parameter set in a previous [Set Sessions Parameter \(9\)](#) function call.

These queued requests are not serviced until the lock is removed. When the lock is removed, the queued requests are processed in first-in-first-out (FIFO) order. EHELLAPI functions not listed are run as if there was no lock. The requesting application unlocks the presentation space window by one of the following methods:

- Disconnecting from the presentation space while still owning the Lock.
- Issuing the [Reset System \(21\)](#) function while still owning the Lock.
- Stopping the application while still owning the Lock.
- Stopping the session.
- Successfully issuing the Lock Presentation Space API with the Unlock option.

Before exiting the application, you should unlock any presentation space windows that have been locked with the Lock Presentation Space API function. If the application exits with outstanding locks, or a [Reset System \(21\)](#), or [Disconnect Presentation](#)

[Space \(2\)](#) function is issued, the locks are released.

It is recommended that applications lock the presentation space only for short periods of time and only when exclusive use of the presentation space is required.

5.2.1.19 Lock Window Services API (61)

The Lock Window Services API function allows the application to obtain or release exclusive control of the presentation space window over other Windows 32-bit applications. While locked, no other application can connect to the presentation space window.

Successful processing of this function with the Lock causes EHLLAPI presentation space window functions requested from other EHLLAPI applications to be queued until the requesting application unlocks the presentation space. Requests from the locking application are processed normally.

3270	5250
Yes	No

Prerequisite Calls

[Connect Window Services \(101\)](#)

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 61	
Data String	See the following table.	
Length	Must be 3	Must be 8
PS Position	NA	

Data String Contents

Byte		Definition
Standard	Enhanced	
1	1	A 1-character presentation space short name (PSID).
	2-4	Reserved.
2	5	One of the following characters: <ul style="list-style-type: none"> • L to lock the API. • U to unlock the API.
3	6	One of the following characters: <ul style="list-style-type: none"> • R to return if the presentation space is already locked by an application. • Q to queue the Lock request if the presentation space is already locked by an application.
	7-8	Reserved.

Return Parameters

Return Code	Explanation
0	The Lock Window Services API function was successful.
1	An incorrect host presentation space short session ID was specified or was not connected.
2	An error was made in specifying parameters.
9	A system error was encountered.
38	Requested function was not complete.
43	The API was already locked by another EHELLAPI application (on LOCK) or API not locked (on UNLOCK).

Remarks

The following EHELLAPI functions are queued when a lock is in effect:

- [Window Status \(104\)](#)
- [Change Switch List Name \(105\)](#)
- [Change PS Window Name \(106\)](#)

These queued requests are not serviced until the lock is removed. When the lock is removed, the queued requests are processed in first-in-first-out (FIFO) order.

The requesting application unlocks the presentation space window by one of the following methods:

- Successfully issuing the Lock Window Services API with the UNLOCK option.
- Disconnecting from the presentation space while still owning the Lock.
- Issuing the [Reset System \(21\)](#) function while still owning the Lock.
- Stopping the application while still owning the Lock.
- Stopping the session.

Before exiting the application, you should Unlock any presentation space windows that have been locked with the Lock Window Services API function. If the application exits with outstanding locks, the subsystem releases the locks.

It is recommended that applications lock the presentation space only for short periods of time and only when exclusive use of the presentation space is required.

5.2.1.20 Pause (18)

The Pause function waits for a specified amount of time. It should be used in place of timing loops to wait for an event to occur. A Pause function can be ended by a host event if a prior [Start Host Notification \(23\)](#) function has been called and the IPAUSE option is selected.

3270	5250
Yes	Yes

Prerequisite Calls

There are no prerequisite calls for this function.

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 18	
Data String	NA	
Length	Contains the pause duration in half-second increments	
PS Position	NA	

Return Parameters

Return Code	Definition
0	The wait duration has expired.
9	An internal system error was encountered. The time results are unpredictable.
26	The host session presentation space or OIA has been updated. Use the Query Host Update (24) function to get more information.

Remarks

1. Selecting the FPAUSE or IPAUSE option using the [Set Session Parameters \(9\)](#) function affects the length of the pause you get when you call this function.
2. The value entered in the calling length parameter is the maximum number of half-second intervals that the Pause function waits. For a pause of 20 seconds, a hex value of 0028 (decimal 40) must be passed in the calling length parameter.
3. If you use the IPAUSE option and the pause value is zero, then the function waits up to 2400 half-second intervals, unless interrupted sooner. If you use the FPAUSE option and the pause value is zero, then the function returns immediately.
4. If you use the IPAUSE option, once a pause has been satisfied by a host event, you should call the [Query Host Update \(24\)](#) function to clear the queue prior to the next Pause function. The Pause function will continue to be satisfied with the pending event until the [Query Host Update \(24\)](#) function is completed.
5. A practical maximum value for the Pause function is 2400. You should not use the Pause function for these kinds of tasks:
 - Delay for very long durations (of several hours, for example).
 - Delay for more than a moderate length of time (20 minutes) before checking the system time-of-day clock and proceeding with your EHLLAPI program execution.
 - With applications requiring a high-resolution timer because the time interval created by a Pause function is approximate.
 - Set the time interval to zero in a loop.
6. IPAUSE set and the interruptible pause allow an EHLLAPI application to determine whether the specified host presentation space (PS) or operator information area (OIA) is updated. The following three functions are used:
 - [Start Host Notification \(23\)](#)
 - [Query Host Update \(24\)](#)
 - [Stop Host Notification \(25\)](#)

By using IPAUSE when the Start function is called, you can make an application wait until the host presentation space or OIA (or both) receives an update. When the receive is completed and the application can issue the Query function to determine the changes, Pause terminates. Then the application issues the [Search](#)

[Presentation Space \(6\)](#) to check whether the expected update occurred.

5.2.1.21 Post Intercept Status (52)

The Post Intercept Status function informs the Personal Communications emulator that a keystroke obtained through the [Get Key \(51\)](#) function was accepted or rejected. When the application rejects a keystroke, the Post Intercept Status function issues a beep.

3270	5250
Yes	Yes

Prerequisite Calls

[Start Keystroke Intercept \(50\)](#)

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 52	
Data String	See the following table	
Length	Must be 2	Must be 8
PS Position	NA	

The calling data string can contain:

Byte		Definition
Standard	Enhanced	
1	1	One of the following values: <ul style="list-style-type: none"> The 1-letter short name of the presentation space. A blank or null indicating a function call for the host-connected presentation space.
	2-4	Reserved
2	5	One of the following characters: <ul style="list-style-type: none"> A for accepted keystroke. R for rejected keystroke.
	6-8	Reserved.

Remarks

Return Code	Explanation
0	The Post Intercept Status function was successful.
1	An incorrect presentation space was specified.
2	An incorrect session option was specified.
8	No prior Start Keystroke Intercept (50) function was called for this presentation space ID.
9	A system error was encountered.

5.2.1.22 Query Additional Field Attribute (45)

The Query Additional Field Attribute function returns additional information about the 5250 field containing the input host presentation space position. This information is returned in the data string parameter in the form of a defined structure.

3270	5250
Yes	Yes

Prerequisite Calls

[Connect Presentation Space \(1\)](#)

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 45.	
Data String	8 bytes long character string.	
Length	8 is implied.	
PS Position	Identifies the target. This can be the PS position of any byte within the target field.	

The calling data string can contain:

Byte	Definition
1-8	Reserved

Return Parameters

This function returns a data string and a return code.

Data String

The function returns the following data string.

Byte	Definition
1-6	Reserved
7-8	Two 8-bit unsigned characters that return: R if field is RTL and L if field is LTR. U if field is upper case and L if field is a normal case field.

Return Code

The following return codes are defined:

Return Code	Explanation
0	The Query Additional Field Attribute was successful.
1	Your program is not currently connected to a host session.
7	The host presentation space position is not valid.

9	No field was found in this position.
24	Field is unformatted.

5.2.1.23 Query Close Intercept (42)

The Query Close Intercept function allows the application to determine if the close option was selected.

3270	5250
Yes	Yes

Prerequisite Calls

[Start Close Intercept \(41\)](#)

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 42	
Data String	See the following table.	
Length	Must be 1	Must be 4
PS Position	NA	

The calling data string can contain:

Byte		Definition
Standard	Enhanced	
1	1	1-character short session ID of the host presentation space, or a blank or null indicating request for querying the host-connected session
	2-4	Reserved

Return Parameters

Return Code	Explanation
0	A close intercept event did not occur.
1	The presentation source was not valid.
2	An error was made in specifying parameters.
8	No prior Start Close Intercept (41) function was called for this host presentation space.
9	A system error occurred.
12	The session stopped.
26	A close intercept occurred since the last query close intercept call.

5.2.1.24 Query Communication Event (81)

The Query Communication Event function lets the EHLAPI program determine whether any communication events have occurred.

3270	5250
Yes	Yes

Prerequisite Calls

[Start Communication Notification \(80\)](#)

Call Parameters

	Enhanced Interface
Function Number	Must be 81
Data String	1-character short name of the host presentation space or a blank or null indicating request for updates to the host-connected presentation space
Length	4 is implied
PS Position	NA

The calling data structure contains these elements:

Byte	Definition
1	A 1-character presentation space short name (PSID)
2-4	Reserved

Return Parameters

Return Code	Definition
0	The function was successful
1	An incorrect PSID was specified
8	No prior call to Start Communication Notification (80) function was called for the PSID
9	A system error was encountered
21	The indicated PSID was connected
22	The Indicated PSID was disconnected

5.2.1.25 Query Cursor Location (7)

The Query Cursor Location function indicates the position of the cursor in the host-connected presentation space by returning the cursor position.

3270	5250
Yes	Yes

Prerequisite Calls

[Connect Presentation Space \(1\)](#)

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 7	
Data String	NA	
Length	NA	
PS Position	NA	

Return Parameters

This function returns a length and a return code.

Length

Host presentation space position of the cursor.

Return Code

The following codes are defined:

Return Code	Explanation
0	The Query Cursor Location function was successful.
1	Your program is not currently connected to a host session.
9	A system error was encountered.

5.2.1.26 Query Field Attribute (14)

The Query Field Attribute function returns the attribute byte of the field containing the input host presentation space position. This information is returned in the returned length parameter.

For the PC/3270, note also that:

- The returned length parameter is set to 0 if the screen is unformatted.
- Attribute bytes are equal to or greater than hex C0.

3270	5250
Yes	Yes

Prerequisite Calls

[Connect Presentation Space \(1\)](#)

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 14.	
Data String	NA.	
Length	NA.	
PS Position	Identifies the target. This can be the PS position of any byte within the	

	target field.	
--	---------------	--

Return Parameters

This function returns a length and a return code.

Length

The attribute value if the screen is formatted, or 0 if the screen is unformatted.

Return Code

The following codes are defined:

Return Code	Explanation
0	The Query Field Attribute was successful.
1	Your program is not currently connected to a host session.
7	The host presentation space position is not valid.
9	A system error was encountered.
24	Attribute byte not found or unformatted host presentation space.

Remarks

The returned field attributes are defined in the following tables. The bit positions are in IBM format with bit 0 as the left most bit in the byte.

3270 Field attribute:

Bit Position	Meaning
0-1	Both = 1, field attribute byte
2	Unprotected/protected 0 = Unprotected data field 1 = Protected field
3	A/N 0 = Alphanumeric data 1 = Numeric data only
4-5	I/SPD 00 = Normal intensity, pen not detectable 01 = Normal intensity, pen detectable 10 = High intensity, pen detectable 11 = Nondisplay, pen not detectable
6	Reserved
7	MDT 0 = Field has not been modified 1 = Field has been modified

5250 field attributes:

Bit Position	Meaning
0	Field attribute flag 0 = Nonfield attribute flag 1 = Field attribute flag
1	Visibility

	0 = Nondisplay 1 = Display
2	Unprotected/protected 0 = Unprotected data field 1 = Protected field
3	Intensity 0 = Normal intensity 1 = High intensity
4-6	Field type 000 = Alphanumeric data: All characters are available 001 = Alphabet only: Uppercase and lowercase, comma, period, hyphen, blank, or Dup key are available 010 = Numeric shift: Automatic shift for number 011 = Numeric data only: 0-9, comma, period, plus, minus, blank, or Dup key are available 101 = Numeric data only: 0-9, or Dup key are available 110 = Magnetic stripe reading device data only 111 = Signed-numeric data: 0-9, plus, minus, or Dup key are available
7	MDT 0 = Field has not been modified 1 = Field has been modified

5.2.1.27 Query Host Update (24)

The Query Host Update function lets the programmed operator determine if the host has updated the host presentation space or OIA because:

- The [Start Host Notification \(23\)](#) function was called (on first call to the Query Host Update function only)
- The previous call to the Query Host Update function (for all calls to the Query Host Update function except the first).

3270	5250
Yes	Yes

Prerequisite Calls

[Start Host Notification \(23\)](#)

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 24	
Data String	1-character short name of the host presentation space, or a blank or null indicating request for updates to host-connected presentation space	
Length	1 is implied	4 is implied
PS Position	NA	

The calling data string can contain:

Byte		Definition
Standard	Enhanced	
1	1	A 1-character presentation space short name (PSID)

	2-4	Reserved
--	-----	----------

Return Parameters

Return Code	Definition
0	No updates have been made since the last call.
1	An incorrect host presentation space was specified.
8	No prior Start Host Notification (23) function was called for the host presentation space ID.
9	A system error was encountered.
21	The OIA was updated.
22	The presentation space was updated.
23	Both the OIA and the host presentation space were updated.
44	Printing has completed in the printer session.

Remarks

The target presentation space must be specified in the data string, even though a connection to the host presentation space is not necessary to check for updates.

5.2.1.28 Query Session Status (22)

The Query Session Status function is used to obtain session-specific information.

3270	5250
Yes	Yes

Prerequisite Calls

There are no prerequisite calls for this function.

Call Parameters

	16-bit	32-bit
Function Number	Must be 22.	
Data String	An 18/20-byte string consisting of a 1-byte short name of the target presentation space plus 17 bytes for returned data. Position 1 can be filled with: <ul style="list-style-type: none"> A blank or a null to indicate a request for the host_connected presentation space. An * (asterisk) to indicate a request for the keyboard-owner presentation space. 	
Length	Must be 18	Must be 20
PS Position	NA	

Return Parameters

This function returns a data string and a return code.

Byte		Definition
Standard	Enhanced	
1	1	A 1-character presentation space short name (PSID)
	2-4	Reserved
2-9	5-12	Session long name (same as profile name; or, if profile not set, same as short name)
10	13	Session Type <ul style="list-style-type: none"> • D 3270 display • E 3270 printer • F 5250 display • G 5250 printer • H ASCII VT
11	14	Session characteristics expressed by a binary number including the following session-characteristics bits <ul style="list-style-type: none"> • Bit 0 EAB 0: Session has the basic attribute. 1: Session has the extended attribute • Bit 1 PSS 0: Session does not support the programmed symbols 1: Session supports the programmed symbols • Bits 2-7 Reserved
12-13	15-16	Number of rows in the host presentation space, expressed as a binary number
14-15	17-18	Number of columns in the host presentation space, expressed as a binary number
16-17	19-20	Host code page expressed as a binary number
18		Reserved

Return Code

The following codes are defined:

Return Code	Explanation
0	The Query Session Status function was successful.
1	An incorrect host presentation space was specified.
2	An incorrect string length was made.
9	A system error was encountered.

Remarks

To use this function, preallocate memory to receive the returned data string parameter. The statements required to preallocate this memory vary depending on the language in which your application is written.

5.2.1.29 Query Sessions (10)

The Query Sessions function returns a 16-byte (12-byte for standard interface) data string describing each host session.

3270	5250
Yes	Yes

Prerequisite Calls

There are no prerequisite calls for this function.

Call Parameters

Function	Description	
	Standard Interface	Enhanced Interface
Function Number	Must be 10	
Data String	Preallocated string of 16n bytes long (12n for 16-bit) (n = number of sessions) or more	
Length	12n bytes	16n bytes
PS Position	NA	

Note:

When the length is not matched to the number of sessions, the return code is 2.

Return Parameters

This function returns a data string, a length, and a return code.

Data String

The returned data string is 16n bytes long (12n for standard interface), where n is the number of host sessions. The descriptors are concatenated into the data string and each session type, and presentation space size of a host session.

The format of each 16-byte (12-byte for standard interface) session descriptor is as follows:

Byte		Definition
Standard	Enhanced	
1	1	A 1-character presentation space short name (PSID)
	2-4	Reserved
2-9	5-12	Session long name (same as profile name; or, if profile not set, same as short name)
10	13	Connection type H=host
	14	Reserved
11-12	15-16	Host presentation space size (this is a binary number and is not in display format). If the session type is a print session, the value is 0.

Length

The number of host sessions started.

Return Code

The following codes are defined:

Return Code	Explanation
0	The Query Sessions function was successful.
2	An incorrect string length was made.
9	A system error was encountered.

Remarks

1. If an application program receives RC=2 or RC=0, the number of the active sessions is returned in the length field. The application program can recognize the minimum string length by this number.
2. The Query Sessions function is affected by the CFGSIZE/NOCFGZISE session option and by the EXTEND_PS/NOEXTEND_PS option.

5.2.1.30 Query System (20)

The Query System function can be used by an EHLAPI application program to determine the level of Personal Communications support and other system-related values. This function returns a string that contains the appropriate system data.

The bytes in this returned string are defined in Return Parameters.

3270	5250
Yes	Yes

Prerequisite Calls

There are no prerequisite calls for this function.

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 20	
Data String	Preallocated string of 35 bytes	36 bytes
Length	Must be 35	Must be 36
PS Position	NA	

Return Parameters

This function returns a data string and a return code.

Data String

A data string of 35 bytes (for 16-bit) or 36 bytes (for 32-bit) is returned. The bytes are defined as follows:

Byte		Definition
Standard	Enhanced	
1	1	EHLAPI version number
2-3	2-3	EHLAPI level number
4-9	4-9	Reserved
10-12	10-12	Reserved
13	13	Hardware base, U=Unable to determine
14	14	Program type

15-16	15-16	Reserved
17-18	17-18	Personal Communications version/level as a 2-byte ASCII value
19	19	Reserved
20-23	20-23	Reserved
24-27	24-27	Reserved
28-29	28-29	Reserved
	30	Reserved
30-31	31-32	NLS type expressed as a 2-byte binary number
33-35	34-36	Reserved

Return Code

The following codes are defined:

Return Code	Explanation
0	The Query System function was successful; data string has been returned.
1	EHLAPI is not loaded. (PC/3270 only)
2	An incorrect string length was specified. (PC/3270 only)
9	A system error was encountered.

Remarks

To use this function, preallocate memory to receive the returned data string parameter.

5.2.1.31 Query Window Coordinates (103)

The Query Window Coordinates function requests the coordinates for the window of a presentation space. The window coordinates are returned in pels.

Note: (0,0) indicates the top-left of the window.

3270	5250
Yes	Yes

Prerequisite Calls

[Connect Window Services \(101\)](#)

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 103	
Data String	1-character short session ID of the host presentation space	
Length	17 is implied	20 is implied
PS Position	NA	

The calling data string can contain:

Byte		Definition
Standard	Enhanced	
1	1	One of the following values: <ul style="list-style-type: none"> • A 1-character presentation space short name (PSID) • A blank or null indicating a function call for the current connection presentation space
	2-4	Reserved
2-17	5-20	Reserved

Return Parameters

This function returns a data string and a return code.

Byte		Definition
Standard	Enhanced	
1	1	One of the following values: <ul style="list-style-type: none"> • A 1-character presentation space short session ID • A blank or null indicating a function call for the current connection presentation space
	2-4	Reserved
2-17	5-20	Four 32-bit unsigned integers that return:
2-5	5-8	XLeft Long integer in pels of the left X coordinate of the rectangular window relative to the desktop window
6-9	9-12	YBottom Long integer in pels of the bottom Y coordinate of the rectangular window relative to the desktop window
10-13	13-15	XRight Long integer in pels of the right X coordinate of the rectangular window relative to the desktop window
14-17	16-20	YTop Long integer in pels of the top Y coordinate of the rectangular window relative to the desktop window

Return Code

The following codes are defined:

Return Code	Explanation
0	The Query Window Coordinates function was successful.
1	Your program was not currently connected to the host session.
9	A system error occurred.
12	The session stopped.

5.2.1.32 Receive File (91)

The Receive File function is used to transfer a file from the host session to the workstation session. It is used the same way as the RECEIVE command is used in the PC/3270. The Receive File function can be called by an EHLLAPI application program.

3270	5250
Yes	Yes

Prerequisite Calls

There are no prerequisite calls for this function.

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 91.	
Data String	Refer to the examples.	
Length	Length, in number of bytes, of the data string. Overridden if in EOT mode.	

Following are examples of the data strings for a single-byte character set (SBSC):

3270 Session

To receive the file from the VM/CMS host system:

```
pc_filename [id:]fn ft [fm] [(option)]
```

To receive the file from the MVS^(TM)/TSO host system:

```
pc_filename[id:]dataset[(member)] [/password] [option]
```

To receive the file from the CICS(R) host system:

```
pc_filename [id:]host_filename [(option)]
```

5250 Session

To receive the file from the eServer i5 or iSeries host system:

```
pc_filename [id:]library file member [option]
```

Following are examples of the data strings for a double-byte character set (DBCS):

3270 Session

To receive the file from the VM/CMS host system:

```
pc_filename [id:]fn ft [fm] [(option)]
```

To receive the file from the MVS/TSO host system:

```
pc_filename [id:]dataset[(member)] [/password] [(option)]
```

To receive the file from the CICS host system:

```
pc_filename [id:]host_filename [(option)]
```

5250 Session

To receive the file from the eServer i5 or iSeries host system:

```
pc_filename [id:]library file member [option]
```

Note:

Parameters within [] are optional. Available options are listed below.

Host System	Common Options
VM/CMS	ASCII, JISCII, CRLF, APPEND, TIME n, CLEAR, NOCLEAR, PROGRESS, QUIET
MVS/TSO	ASCII, JISCII, CRLF, APPEND, TIME (n), CLEAR, NOCLEAR, PROGRESS, QUIET, AVBLOCK TRACKS CYLINDERS
CICS	ASCII, JISCII, CRLF, NOCRLF, BINARY, TIME n, CLEAR, NOCLEAR, PROGRESS, QUIET
i5/OS(TM) or OS/400(R)	ASCII, JISCII, CRLF, APPEND, TIME n, CLEAR, NOCLEAR, PROGRESS, QUIET

Note: JISCII is valid in a DBCS session for Japan only and ASCII is valid for all other SBCS and DBCS sessions. Other options specified will be passed to the host transfer program. The file transfer program on the host side either uses them, ignores them, or returns an error. Consult the host transfer program documentation to see a complete list of the options supported.

Return Parameters

Return Code	Explanation
2	Parameter error or you have specified a length that is too long (more than 255 bytes) for the EHLLAPI buffer. The file transfer was unsuccessful.
3	File transfer complete.
4	File transfer complete with segmented records.
9	A system error was encountered.
27	File transfer terminated because of either a Cancel button or the timeout set by the Set Session Parameter (9) function.
101	File transfer was successful (transfer to/from CICS).

If you receive return code 2 or 9, there is a problem with the system or with the way you specified your data string.

Other return codes can also be received, which relate to message numbers generated by the host transfer program. For transfers to a CICS host transfer program, subtract 100 from the return code to give you the numeric portion of the message. For example, a return code of 101 would mean that the message number INW0001 was issued by the host. For other host transfer programs, just use the return code as the numerical part of the message. For example, a return of 34 would mean that message TRANS34 was issued by the host transfer program. The documentation for your host transfer program should give more information about the meanings of the specific messages.

Operating system error codes reported by EHLLAPI are greater than 300. To determine the error code, subtract 300 and refer to the operating system documentation for return codes.

Remarks

- Four sets of parameters under the [Set Session Parameters \(9\)](#) function are related to this function. They are the STRLEN/STREOT, EOT=c, QUIET/NOQUIET and the

TIMEOUT=c/TIMEOUT=0 session options.

2. If no path is specified when the Receive File function is executed, the received file is stored in the current subdirectory, which is the directory in which your application is running.

5.2.1.33 Release (12)

The Release function unlocks the keyboard that is associated with the host presentation space reserved using the [Reserve \(11\)](#) function.

3270	5250
Yes	Yes

Prerequisite Calls

[Connect Presentation Space \(1\)](#)

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 12	
Data String	NA	
Length	NA	
PS Position	NA	

Return Parameters

Return Code	Explanation
0	The Release function was successful.
1	Your program is not connected to a host session.
9	A system error was encountered.

Remarks

If you do not Release a host presentation space reserved by using the [Reserve \(11\)](#) function, you are locked out of that session until you call the [Reset System \(21\)](#) function, you call the [Disconnect Presentation Space \(2\)](#) function, or you terminate the EHLAPI application program.

5.2.1.34 Reserve (11)

The Reserve function locks the keyboard that is associated with the host-connected presentation space to block input from the terminal operator.

The reserved host presentation space remains locked until one of the following occurs:

- Connect (1) function is executed to a new session.

- Disconnect Presentation Space (2) function is executed.
- Release (12) function is executed.
- Reset System (21) function is executed.
- Start Keystroke Intercept (50) function is executed.
- EHLLAPI application program is terminated.

3270	5250
Yes	Yes

Prerequisite Calls

[Connect Presentation Space \(1\)](#)

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 11	
Data String	NA	
Length	NA	
PS Position	NA	

Return Parameters

Return Code	Explanation
0	The Reserve function was successful.
1	Your program is not connected to a host session.
5	Presentation space cannot be used.
9	A system error was encountered.

Remarks

1. If your EHLLAPI application program is sending a series of transactions to the host, you might need to prevent the user from gaining access to that session until your application processing is complete.
2. The keyboard input that a user makes while the keyboard is locked by this function is enqueued and processed after the session is terminated.
3. This function locks both the mouse and the keyboard input. The application program must unlock the presentation space to enable either the mouse or the keyboard input.

5.2.1.35 Reset System (21)

The Reset System function reinitializes EHLLAPI to its starting state. The session parameter options are reset to their defaults. Event notification is stopped. The reserved host session is released. The host presentation space is disconnected. Keystroke intercept is disabled.

You can use the Reset System function during initialization or at program termination to reset the system to a known initial condition.

3270	5250
Yes	Yes

Prerequisite Calls

There are no prerequisite calls for this function.

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 21	
Data String	NA	
Length	NA	
PS Position	NA	

Return Parameters

Return Code	Definition
0	The Reset System function was successful.
1	EHLAPI is not loaded.
9	A system error was encountered.

Remarks

For the PC/3270, this function can be used to check whether EHLAPI is loaded. Place a call to this function at the start of your application and check for a return code of 1.

5.2.1.36 Search Field (30)

The Search Field function examines a field within the connected host presentation space for the occurrence of a specified string. If the target string is found, this function returns the decimal position of the string numbered from the beginning of the host presentation space. (For example, in a 24-row by 80-column presentation space, the row 1, column 1 position is numbered 1 and the row 5, column 1 position is numbered 321.)

This function can be used to search either protected or unprotected fields, but only in a field-formatted host presentation space.

Note:

If the field at the end of the host presentation space wraps, wrapping occurs when the end of the presentation space is reached.

3270	5250
Yes	Yes

Prerequisite Calls

[Connect Presentation Space \(1\)](#)

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 30.	
Data String	Target string for search.	
Length	Length of the target data string. Overridden in EOT mode.	
PS Position	Identifies the target field. For SRCHALL, this can be the PS position of any byte within the target field. For SRCHFROM, it is the beginning point of the search for SRCHFRWD or the ending point of the search for SRCHBKWD. See remark 3.	

Return Parameters

This function returns a length and a return code.

Length

The following codes are defined:

Length	Explanation
= 0	The string was not found.
> 0	The string was found at the indicated host presentation space position.

Return Code

The following codes are defined:

Return Code	Explanation
0	The Search Field function was successful.
1	Your program is not connected to a host session.
2	Parameter error. Either the string length was zero, or EOT mode was specified but no EOT character was found in calling data string.
7	The host presentation space position is not valid.
9	A system error was encountered.
24	The search string was not found, or the host presentation space was unformatted.

Remarks

1. Four sets of parameters under the [Set Session Parameters \(9\)](#) function are related to this function. They are the SRCHALL/SRCHFROM, STRLEN/STREOT, SRCHFRWD/SRCHBKWD, and the EOT=c session options.
2. You can use the Set Session Parameters (9) function to determine whether your searches proceed forward (SRCHFRWD) or backward (SRCHBKWD) in a field.
3. The Search Field function normally checks the entire field (SRCHALL default mode). However, you can use the function 9 to specify SRCHFROM. In this mode, the calling PS position parameter does more than identify the target field. It also provides a beginning or ending point for the search.
 - If the SRCHFRWD option is in effect, the search for the designated string begins at the specified PS position and proceeds toward the end of the field.
 - If the SRCHBKWD option is in effect, the search for the designated string begins at the end of the field and proceeds backward toward the specified PS position. If the target string is not found, the search ends at the PS position specified in the calling PS position parameter.
4. DBCS Only: If the start position of the specified search function is the second byte in a double-byte character, the search is started from the next character for SRCHFRWD and from the character for SRCHBKWD. If the last character of the specified string is the first byte of a double-byte character, the character is not searched for.
The search ignores a pair of SO and SI in the presentation space. When you search a double-byte control character, put SO (X'0E') before the character and SI (X'0F') after it. For example, X'0E000C0F' in the data string is treated as a double-byte character FF (X'000C').

5.2.1.37 Search Presentation Space (6)

The Search Presentation Space function lets your EHLLAPI program examine the host presentation space for the occurrence of a specified string.

3270	5250
Yes	Yes

Prerequisite Calls

[Connect Presentation Space \(1\)](#)

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 6.	
Data String	Target string for search.	
Length	Length of the target data string. Overridden in EOT mode.	
PS Position	Position within the host presentation space where the search is to begin (SRCHFRWD option) or to end (SRCHBKWD option). Overridden in SRCHALL (default) mode.	

Return Parameters

This function returns a length and a return code.

Length

The following codes are defined:

Length	Explanation
= 0	The string was not found.
> 0	The string was found at the indicated host presentation space position.

Return Code

The following codes are defined:

Return Code	Explanation
0	The Search Presentation Space function was successful.
1	Your program is not connected to a host session.
2	An error was made in specifying parameters.
7	The host presentation space position is not valid.
9	A system error was encountered.
24	The search string was not found.

Remarks

- Four sets of parameters under the [Set Session Parameters \(9\)](#) function are related to this function. They are the SRCHALL/SRCHFROM, STRLEN/STREOT, SRCHFRWD/SRCHBKWD, and the EOT=c session options.
- You can use the [Set Session Parameters \(9\)](#) function to specify SRCHBKWD. When this option is in effect, the search operation locates the last occurrence of the string.
- The Search Presentation Space function normally checks the entire host presentation space. However, you can use the [Set Session Parameters \(9\)](#) function to specify SRCHFROM. In this mode, the calling PS position parameter specifies a beginning or ending point for the search.
 - If the SRCHFRWD option is in effect, the search for the designated string begins at the specified PS position and proceeds toward the end of the host presentation space.
 - If the SRCHBKWD option is in effect, the search for the designated string begins at the end of the PS and proceeds backward toward the specified PS position. If the target string is not found, the search ends at the PS position specified in the calling PS position parameter.
- The SRCHFROM option is also useful if you are looking for a keyword that might occur more than once in the host presentation space.
- The Search Presentation Space function is useful in determining when the host presentation space is available. If your EHELLAPI application is expecting a specific prompt or message before sending data, the Search Presentation Space function allows you to check for a prompt message before continuing.
- DBCS Only: If the start position of the specified search function is the second byte in a double-byte character, the search is started from the next character for SRCHFRWD and from the character for SRCHBKWD. If the last character of the specified string is the first byte of a double-byte character, the character is not

searched for.

The search ignores a pair of SO and SI in the presentation space. When you search a double-byte control character, put SO (X'0E') before the character and SI (X'0F') after it. For example, X'0E000C0F' in the data string is treated as a double-byte character FF (X'000C').

5.2.1.38 Send File (90)

The Send File function is used to transfer a file from the workstation session where EHLAPI is running to a host session.

3270	5250
Yes	Yes

Prerequisite Calls

There are no prerequisite calls for this function.

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 90.	
Data String	Refer to the examples.	
Length	Length of the target data string. Overridden in EOT mode.	
PS Position	Must be 0.	

Following are examples of the data strings for SBCS

3270 Session

To send the file to the VM/CMS host system:

```
pc_filename [id:]fn ft [fm] [(option)]
```

To send the file to the MVS/TSO host system:

```
pc_filename [id:]dataset[(member)] [/password] [option]
```

To send the file to the CICS host system:

```
pc_filename [id:]host_filename [(option)]
```

5259 Session

To send the file to the eServer i5 or iSeries host system:

```
pc_filename [id:]library file member [option]
```

Following are examples of the data strings for DBCS:

3270 Session

To send the file to the VM/CMS host system:

`pc_filename [id:]fn ft [fm] [(option)]`

To send the file to the MVS/TSO host system:

`pc_filename [id:]dataset[(member)] [/password] [(option)]`

To send the file to the CICS host system:

`pc_filename [id:]host_filename [(option)]`

5250 Session

To send the file to the eServer i5 or iSeries host system:

`pc_filename [id:]library file member [option]`

Note:

Parameters within [] are optional. Available options are listed below.

Host System	Common Options
VM/CMS	ASCII, JISCII, CRLF, APPEND, LRECL n, RECFM v f, TIME n, CLEAR, NOCLEAR, PROGRESS, QUIET
MVS/TSO	ASCII, JISCII, CRLF, APPEND, LRECL (n), RECFM (v f u), TIME (n), CLEAR, NOCLEAR, PROGRESS, QUIET, BLKSIZE (n), SPACE (n[,m]), AVBLOCK TRACKS CYLINDERS
CICS	ASCII, JISCII, CRLF, BINARY, TIME n, CLEAR, NOCLEAR, PROGRESS, QUIET
i5/OS or OS/400	ASCII, JISCII, CRLF, APPEND, SRC, LRECL n, TIME n, CLEAR, NOCLEAR, PROGRESS, QUIET

Note: JISCII is valid in a DBCS session for Japan only and ASCII is valid for all other SBCS and DBCS sessions. Note: Time, if specified, overrides the value in Set Session parameters. Note: Other options specified will be passed to the host transfer program. The file transfer program on the host side either uses them, ignores them, or returns an error. Consult the host transfer program documentation to see a complete list of the options supported.

Return Parameters

Return Code	Explanation
2	Parameter error or you have specified a length that is too long (more than 255 bytes) for the EHLAPI buffer. The file transfer was unsuccessful.
3	File transfer complete.
4	File transfer complete with segmented records.
5	Workstation file name is not valid or not found. File transfer was canceled.
9	A system error was encountered.
27	File transfer terminated because of either a Cancel button or the timeout set by the Set Session Parameter (9) function.
101	File transfer was successful (transfer to/from CICS).

If you receive return code 2 or 9, there is a problem with the system or with the way you specified your data string.

Other return codes can also be received which relate to message numbers generated by the host transfer program. For transfers to a CICS host transfer program, subtract 100 from the return code to give you the numeric portion of the message. For example, a return code of 101 would mean that the message number INW0001 was issued by the host. For other host transfer programs, just use the return code as the numerical part of the message. For example, a return of 34 would mean that message TRANS34 was issued by the host transfer program. The documentation for your host transfer program should give more information about the meanings of the specific messages.

Operating system error codes reported by EHLLAPI are greater than 300. To determine the error code, subtract 300 and refer to the operating system documentation for return codes.

Remarks

Four sets of parameters under the [Set Session Parameters \(9\)](#) function are related to this function. They are the QUIET/NOQUIET, STRLEN/STREOT, TIMEOUT=c/TIMEOUT=0, and the EOT=c session options.

5.2.1.39 Send Key (3)

The Send Key function is used to send either a keystroke or a string of keystrokes to the host presentation space.

You define the string of keystrokes to be sent with the calling data string parameter. The keystrokes appear to the target session as though they were entered by the terminal operator. You can also send all attention identifier (AID) keys such as Enter and so on. All host fields that are input protected or are numeric only must be treated accordingly.

3270	5250
Yes	Yes

Prerequisite Calls

[Connect Presentation Space \(1\)](#)

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 3.	
Data String	A string of keystrokes, maximum 255. Uppercase and lowercase ASCII characters are represented literally. Function keys and shifted function keys are represented by mnemonics.	

	See Keyboard Mnemonics.	
Length	Length of the source data string. Overridden if in EOT mode.	
PS Position	NA	

Return Parameters

Return Code	Explanation
0	The keystrokes were sent; status is normal.
1	Your program is not connected to a host session.
2	An incorrect parameter was passed to EHELLAPI.
4	The host session was busy; all of the keystrokes could not be sent.
5	Input to the target session was inhibited or rejected; all of the keystrokes could not be sent.
9	A system error was encountered.

Remarks

- The parameters under the Set Session Parameters (9) function are related to this function. They are the AUTORESET/NORESET, STRLEN/STREOT, EOT=c, ESC=c, and RETRY/NORETRY session options.
- Keystrokes cannot be sent to the host session when the keyboard is locked or busy. You can check this condition with the [Wait \(4\)](#) function.
- If the host is busy, input might be rejected.
- The length of the data string must be explicitly defined by the default length parameter, but it can be defined implicitly by the EOT=c option of the [Set Session Parameters \(9\)](#) function.
When explicitly defining length, the value for the length parameter passed by the application must be calculated. For this calculation, allow 2 bytes for compound keystrokes such as @E and allow 4 bytes for compound keystrokes such as @A@C.
- To send special control keys, a compound character coding scheme is used. In this coding scheme, one keystroke is represented by a sequence of two to four ASCII characters. The first and third character are always the escape character. The second and fourth character are always a keycode.
To send the sequence LOGON ABCDE followed by the Enter key, you would code the string LOGON ABCDE@E. A complete list of these keycodes is represented in Keyboard Mnemonics.
This compound coding technique allows an ASCII string representation of all necessary keystroke codes without requiring the use of complex hexadecimal key codes.
The default escape character is @. The value of the escape character can be changed to any other character with the ESC=c option of the [Set Session Parameters \(9\)](#) function.
- Users needing higher levels of performance should use the [Copy String to Field \(33\)](#) or [Copy String to Presentation Space \(15\)](#) function rather than send keystrokes with the [Send Key \(3\)](#) function. But remember, only the [Send Key \(3\)](#) function can send the special control keys.
- Refer to [Set Session Parameters \(9\)](#) session option 10 (NORESET option) to improve the performance of this function.
Unless NORESET is required, the reset mnemonic is added to the keystroke strings as a prefix. Therefore, all resettable status except input inhibit are reset.

- The NORESET option is not the same as the [Reset System \(21\)](#) function.
8. The keystroke strings, including the AID key, are sent to the host via multiple paths. Each path sends the strings before the first AID key (or including the AID key). EHLLAPI adjusts the string length and the start position of each path. For a host application program, any keystroke might be lost by the AID key process. Therefore, you should not send a keystroke list that includes plural AID keys.
 9. During the @P (Print) or @A@T (Print Presentation Space) process, all requests that update the presentation space are rejected. If the presentation space is busy or the interruption request occurs during the print request, the mnemonic @A@R (Device Reset - Cancel to print the Presentation Space) cancels the request and resets the status.

5.2.1.40 Set Cursor (40)

The Set Cursor function is used to set the position of the cursor within the host presentation space. Before using the Set Cursor function, a workstation application must be connected to the host presentation space.

3270	5250
Yes	Yes

Prerequisite Calls

[Connect Presentation Space \(1\)](#)

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 40	
Data String	NA	
Length	NA	
PS Position	Desired cursor position in the connected host presentation space	

Return Parameters

Return Code	Explanation
0	Cursor was successfully located at the specified position.
1	Your program is not connected to a host session.
4	The session is busy.
7	A cursor location less than 1 or greater than the size of the connected host presentation space was specified.
9	A system error occurred.

Remarks

DBCS Only: If the specified cursor is the second byte of the double-byte character, the cursor moves to the first byte of the character and an error code is not returned.

5.2.1.41 Set Session Parameters (9)

The Set Session Parameters function lets you change certain default session options in EHLLAPI for all sessions. When EHLLAPI is loaded, the default settings for session options are as indicated by the underscored entries in the tables that appear in Session Options . Any, some, or all of these settings can be changed by including the desired option in the calling data string as explained below. Specified settings remain in effect until:

- Changed by a subsequent Set Session Parameters (9) function that specifies a new value.
- The [Reset System \(21\)](#) function is executed.
- The EHLLAPI application program is terminated.

3270	5250
Yes	Yes

The following table lists those EHLLAPI functions that are affected by session options. Functions not listed in the table are not affected by any of the session options. Session options that affect each function are indicated by corresponding entries in the "See Items" column. These entries are indexed to the list that follows Call Parameters.

Function Number	Function Name	See Items
1	Connect Presentation Space	11, 23, 24
3	Send Key	1, 2, 9, 10, 19
4	Wait	12
5	Copy Presentation Space	5, 13, 14, 15, 17, 20, 21, 22
6	Search Presentation Space	1, 2, 3, 4
8	Copy Presentation Space to String	5, 13, 14, 15, 17, 20, 21, 22
10	Query Sessions	16, 22
15	Copy String to Presentation Space	1, 2, 13, 14, 18, 20, 21, 22
18	Pause	6
30	Search Field	1, 2, 3, 4, 22
33	Copy String to Field	1, 2, 13, 14, 18, 20, 21, 22
34	Copy Field to String	5, 13, 14, 17, 20, 21, 22
51	Get Key	9, 12
90	Send File	1, 2, 7, 8
91	Receive File	1, 2, 7, 8
101	Connect Window Services	23, 24

Note: Items 20 and 21 in this table are for DBCS only

Prerequisite Calls

There are no prerequisite calls for this function.

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 9.	
Data String	String containing the desired values of those session options that are to be	

	changed. The data string can contain any of the values in the tables of Session Options. The values should be placed on the data string line, separated by commas or blanks. The sets of parameters are explained in terms of the functions they affect.	
Length	Explicit length of the source data string (the STREOT option is not allowed).	
PS Position	NA.	

Session Options

The following tables show the session options. The default is underlined.

- The values in the following table determine how the data string length is defined for functions [Send Key \(3\)](#), [Search Presentation Space \(6\)](#), [Copy String to Presentation Space \(15\)](#), [Search Field \(30\)](#), [Copy String to Field \(33\)](#), [Send File \(90\)](#), and [Receive File \(91\)](#).

Value	Explanation
<u>STRLEN</u>	An explicit length is passed for all strings.
STREOT	Lengths are not explicitly coded. Calling (source) data strings are terminated with an EOT character.

- The statement in the following table is used to specify the character that is used as the end-of-text (EOT) delimiter in the calling (source) data string for EHLAPI functions [Send Key \(3\)](#), [Search Presentation Space \(6\)](#), [Copy String to Presentation Space \(15\)](#), [Search Field \(30\)](#), [Copy String to Field \(33\)](#), [Send File \(90\)](#), and [Receive File \(91\)](#).

Value	Explanation
EOT=c	Allows you to specify the EOT character for string terminators (in STREOT mode). Binary zero is the default. Do not leave a blank after the equal sign.

To be valid, c must be entered as a 1-byte string literal character with no preceding blanks. The EOT character specified by this statement is used to determine the length of a calling data string only when the STREOT option (see item 1) is in effect.

- The values in the following table affect the [Search Presentation Space \(6\)](#) and [Search Field \(30\)](#) search functions.

Value	Explanation
<u>SRCHALL</u>	The Search Presentation Space (6) function and Search Field (30) function scan the entire host presentation space or field.
SRCHFROM	The Search Presentation Space (6) function and Search Field (30) function start from a specified PS position (for SRCHFRWD) or end at a specified PS position (for SRCHBKWD).

- The values in the following table affect the [Search Presentation Space \(6\)](#) and [Search Field \(30\)](#) search functions. They determine the direction for the search.

Value	Explanation
-------	-------------

SRCHFRWD	The Search Presentation Space (6) function and Search Field (30) function perform in an ascending direction.
SRCHBKWD	The Search Presentation Space (6) function and Search Field (30) function perform in a descending direction. A search is satisfied if the first character of the requested string starts within the bounds specified for the search.

5. The values in the following table determine how attribute bytes are treated for functions [Copy Presentation Space \(5\)](#), [Copy Presentation Space to String \(8\)](#), and [Copy Field to String \(34\)](#).

Value	Explanation
NOATTRB	Convert all unknown values to blanks.
ATTRB	Pass back all codes that do not have an ASCII equivalent as their original values.
NULLATTRB	Convert all field attributes to null characters.

6. The values in the following table affect the [Pause \(18\)](#) function.

Value	Explanation
FPAUSE	A full-duration pause lasts for however long you specified in the Pause (18) function.
IPAUSE	Interruptible pause. After the Start Host Notification (23) function is executed, a host event satisfies a pause.

7. The values in the following table determine whether messages generated by file transfer functions [Send File \(90\)](#) and [Receive File \(91\)](#) are displayed.

Value	Explanation
NOQUIET	SEND and RECEIVE messages are displayed.
QUIET	SEND and RECEIVE messages are not displayed.

8. The statements in the following table determine how long EHLLAPI waits before it automatically issues a Cancel during execution of file transfer functions [Send File \(90\)](#) and [Receive File \(91\)](#). To be valid, c must be an Arabic number 0-9 or a capital letter J-N and must not be preceded by a blank.

Value	Explanation
TIMEOUT=0	A Cancel is automatically issued following a 20-second (approximate) delay.
TIMEOUT=c	A Cancel is automatically issued following a specified delay. A 1-character indicator from the table below tells the application how many 30-second cycles it should accept before issuing a Cancel itself. Character value (in minutes) 1 0.5 2 1.0 3 1.5 4 2.0 5 2.5 6 3.0 7 3.5 8 4.0 9 4.5 J 5.0 K 5.5 L 6.0 M 6.5 N 7.0

9. The statement in the following table is used to define the escape character for keystroke mnemonics. This session option affects functions [Send Key \(3\)](#) and [Get](#)

[Key \(51\)](#). The value of c must be entered as a 1-byte literal character string with no preceding blanks.

Value	Explanation
ESC=c	Specifies the escape character for keystroke mnemonics (@ is the default). Do not leave a blank after the equal sign. A blank is not a valid escape character.

10. The values in the following table determine whether EHLAPI automatically precedes strings sent using the [Send Key \(3\)](#) function with a reset.

Value	Explanation
AUTORESET	EHLAPI attempts to reset all inhibited conditions by prefixing all strings of keys sent using the Send Key (3) function with a reset.
NORESET	Do not AUTORESET.

11. The values in the following table affect the manner in which the [Connect Presentation Space \(1\)](#) command function.

Value	Explanation
CONLOG	Establishes a logical connection between the workstation session and a host session. During Connect, does not jump to the requested presentation space.
CONPHYS	Establishes a physical connection between the workstation session and a host session. During Connect, jumps to the requested presentation space.

12. The values in the following table affect the [Wait \(4\)](#) function and [Get Key \(51\)](#) function. For each value, there are two different effects, one for each function.

Value	Explanation
TWAIT	For the Wait (4) function, waits up to a minute before timing out on XCLOCK (X []) or XSYSTEM. For the Get Key (51) function, does not return control to your EHLAPI application program until it has intercepted a key (normal or AID key based on the option specified under the Start Keystroke Intercept (50) function).
LWAIT	For the Wait (4) function, waits until XCLOCK (X [])/XSYSTEM clears. This option is not recommended, because control does not return to your application until the host is available. For the Get Key (51) function, does not return control to your EHLAPI application program until it has intercepted a key (normal or AID key based on the option specified under the Start Keystroke Intercept (50) function).
NWAIT	For the Wait (4) function, checks status and returns immediately (no wait). For the Get Key (51) function, returns return code 25 (keystrokes not available) in the fourth parameter if nothing is queued matching the option specified under the Start Keystroke Intercept (50) function.

Note:

Use of NWAIT is recommended.

13. The values in the following table affect [Copy Presentation Space \(5\)](#), [Copy Presentation Space to String \(8\)](#), [Copy String to Presentation Space \(15\)](#), [Copy String to Field \(33\)](#), and [Copy Field to String \(34\)](#). Extended attribute bytes (EAB) include extended character attributes and extended field attributes.

Value	Explanation
NOEAB	Pass data only, no EABs.
EAB	Pass the presentation space data with extended attribute bytes. For each character that appears on the screen, 2 bytes of data are passed. Therefore, a buffer twice the size of the presentation space must be preallocated; for example $2 \times 1920 = 3840$ for a 24-row by 80-column presentation space. Extended attributes for a string of characters may be reported as attributes of the field byte, rather than as attributes of each individual character in the field. In this case, to tell if a particular character

or set of characters on a screen is underscored, do a CopyPStoString specifying the position of the field attribute byte (the byte before the field that is displayed on the screen) to get the EAB information that applies to all of the characters in that field.
--

Note:

When using EHLLAPI Copy PS to String, text is copied which should be invisible to the operator. Use the EHLLAPI Set Session Parameters function to set the NODISPLAY option to determine if there is hidden data. This causes EHLLAPI to return nondisplay fields as nulls. Another common procedure for hiding data is to set the foreground and background colors the same (BLACK, for instance) so the text is displayed, but not visible to the human operator. The only way for your application to detect this is to use the EAB and XLATE session parameters and then copying the PS. The foreground/background color of each position is returned and you can determine which characters are invisible.

14. The values in the following table affect [Copy Presentation Space \(5\)](#), [Copy Presentation Space to String \(8\)](#), [Copy String to Presentation Space \(15\)](#), [Copy String to Field \(33\)](#), and [Copy Field to String \(34\)](#).

Value	Explanation
NOXLATE	EABs are not translated.
XLATE	EABs are translated to the PC color graphics adapter (CGA) format.

15. The values in the following table affect [Copy Presentation Space \(5\)](#) and [Copy Presentation Space to String \(8\)](#) if NOATTRB and NOEAB are specified.

Value	Explanation
BLANK	Convert all unknown values to X'20'.
NOBLANK	Convert all unknown values to X'00'.

16. The values in the following table affect the presentation space size that is returned by the [Query Sessions \(10\)](#).

Value	Explanation
CFGSIZE	Returns the configured size of the connected presentation space. This option ignores any override of the configured size by the host.
NOCFGSIZE	Returns the current size of the connected presentation space.

17. The values in the following table affect [Copy Presentation Space \(5\)](#), [Copy Presentation Space to String \(8\)](#), and [Copy Field to String \(34\)](#).

Value	Explanation
DISPLAY	Copy nondisplay fields in the presentation space to the target buffer area in the same manner as display fields. Current applications function normally.
NODISPLAY	Do not copy nondisplay fields in the presentation space to the target buffer area. Copy the nondisplay fields to the target buffer as a string of null characters. This allows applications to display the copied buffers in the presentation widow without displaying confidential information, such as passwords.

18. The values in the following table affect [Copy String to Presentation Space \(15\)](#) and [Copy String to Field \(33\)](#).

Value	Explanation
NOPUTEAB	EAB (or EAD for DBCS) is not contained in the data string of Copy String to Presentation Space or

	Copy String to Field.
PUTEAB	EAB is contained with character data in the data string of Copy String to Presentation Space or Copy String to Field.

19. The values in the following table affect the [Send Key \(3\)](#) function. Keystrokes are not processed if the keyboard is blocked or in use. The options determine whether the function tries to resend the keystrokes until a 4-minute timeout occurs or if the function returns immediately after determining the keyboard is blocked or in use.

Value	Explanation
RETRY	Continues to attempt to send keystrokes until they are sent or until a 4-minute timeout occurs.
NORETRY	Returns immediately after determining the keyboard is blocked or in use.

20. DBCS Only: The values in the following table affect [Copy Presentation Space \(5\)](#), [Copy Presentation Space to String \(8\)](#), [Copy String to Presentation Space \(15\)](#), [Copy String to Field \(33\)](#), and [Copy Field to String \(34\)](#).

Value	Explanation
NOEAD	DBCS attribute characters are not passed.
EAD	Pass the presentation space data and two attribute characters for the double-byte character set (DBCS). (Users receive 2 bytes for each character other than the data. Therefore, a buffer twice the size of the presentation space must be preallocated.)

21. DBCS Only: The values in the following table affect [Copy Presentation Space \(5\)](#), [Copy Presentation Space to String \(8\)](#), [Copy String to Presentation Space \(15\)](#), [Copy String to Field \(33\)](#), and [Copy Field to String \(34\)](#).

Value	Explanation
<u>NO</u> SO	Pass the presentation space data except Shift-in (SI) and Shift-out (SO) control characters.
SO	Pass the presentation space data including translated SI control character to X'0E' and SO control character to X'0F'. The allocated buffer size depends on the length of the stored data.
SPACESO	Pass the presentation space data including translated SI and SO control characters to X'20' (blank). The allocated buffer size depends on the length of the stored data.

22. The values in the following table affect [Copy Presentation Space \(5\)](#), [Copy Presentation Space to String \(8\)](#), [Copy String to Presentation Space \(15\)](#), [Copy String to Field \(33\)](#), [Copy Field to String \(34\)](#), [Search Field \(30\)](#) and [Query Sessions \(10\)](#).

Value	Explanation
<u>EXTEND</u> _PS	5250 emulation supports a presentation space of 24 rows by 80 columns. In some instances, Communication Manager 5250 emulation displays a 25th row. This occurs when either an error message from the host is displayed or when the operator selects the SysReq key.
NO <u>EXTEND</u> _PS	The presentation space is not extended when the above condition occurs. This is the default value.

23. The values in the following table affect the [Connect Presentation Space \(1\)](#) and [Connect Window Services \(101\)](#) functions. The options specify whether an application can or will share the presentation space to which it is connected with another application. Only one of the following values can be specified with each Set Session Parameter call.

Value	Explanation
<u>SUPER_WRITE</u>	The application allows other applications that allow sharing and have write access permissions to concurrently connect to the same presentation space. The originating application performs

	supervisory-type functions but does not create errors for other applications that share the presentation space.
WRITE_SUPER	The application requires write access and allows only supervisory application to concurrently connect to its presentation space. This is the default value.
WRITE_WRITE	The application requires write access and allows partner or other applications with predictable behavior to share the presentation space.
WRITE_READ	The application requires write access and allows other applications that perform read-only functions to share the presentation space. The application is also allowed to copy the presentation space and perform other read-only operations as usual.
WRITE_NONE	The application has exclusive use of the presentation space. No other applications are allowed to share the presentation space, including supervisory applications. The application is allowed to copy the presentation space and perform read-only operations as usual.
READ_WRITE	The application requires only read access to monitor the presentation space and allows other applications that perform read or write, or both, functions to share the presentation space. The application is also allowed to copy the presentation space and perform other read-only operations as usual.

24. The values in the following table allow applications that have presentation space sharing requirements to limit the sharing to a partner application (an application that was developed to work with it).

Value	Explanation
<u>NOKEY</u>	Allows the application to be compatible with existing applications that do not specify the KEY parameter.
KEY\$nnnnnnn	Uses a keyword to restrict sharing access to the presentation space that it supports. The keyword must be exactly 8 bytes in length.

Return Parameters

This function returns a length and a return code.

Length

Number of valid session parameters that are set.

Return Code

The following codes are defined:

Return Code	Explanation
0	The session parameters have been set.
2	One or more parameters were not valid.
9	A system error was encountered.

5.2.1.42 Start Close Intercept (41)

The Start Close Intercept function allows the application to intercept close requests generated when a user selects the close option from the emulator session window. This function intercepts the close request and discards it until a [Stop Close Intercept \(43\)](#) function is requested.

After using this function, your application program can use the [Query Close Intercept \(42\)](#) function to determine when a close request has occurred.

3270	5250
------	------

Yes	Yes
-----	-----

Prerequisite Calls

There are no prerequisite calls for this function.

Call Parameters

Byte	Definition	
	Standard Interface	Enhanced Interface
Function Number	Must be 41	
Data String	See the following table	
Length	5 or 6	Must be 12
PS Position	NA	

The data string contains the following items.

Byte		Definition
Standard	Enhanced	
1	1	A 1-character presentation space short name (PSID).
	2-4	Reserved.
4-5		The data in these positions is ignored by EHLLAPI. However, no error is caused if the migrating program has data in these positions. This data is accepted to provide compatibility with migrating applications.
6	5	Specify M to request asynchronous message mode (Windows only).
	6-8	Reserved.
2-3	9-12	When M is specified in position 5 (6 for 16-bit), the window handle of the window that receives the message should be set. The message is a return value of RegisterWindowMessage (PCSHLL) (not equal 0).

Return Parameters

This function returns a data string and a return code.

Data String

If asynchronous message mode is not specified in position 5 (6 for standard interface) and the function is completed successfully, the following data string is returned.

Byte		Definition
Standard	Enhanced	
1	1	A 1-character presentation space short name (PSID).
	2-8	Reserved.
	9-12	4 byte value in which the event object address is returned by EHLLAPI. The application can wait for this event object. (32-bit only).

Data String

If M (asynchronous message mode) is specified in position 5 (6 for standard interface) and the function is completed successfully, the following data string is returned.

Byte		Definition
Standard	Enhanced	
1	1	A 1-character presentation space short name (PSID)
	2-8	Reserved
2-3	9-10	Task ID of asynchronous message mode

Note:

If a user selects the close option, an application window receives a message. The message is a return value of RegisterWindowMessage (PCSHLL). The wParam parameter will contain the Task ID returned by this function call. The HIWORD of the lParam parameter will contain the Return Code 26, which shows a close intercept occurred, and the LOWORD of the lParam parameter will contain the function number 41.

Return Code

The following codes are defined:

Return Code	Explanation
0	The Start Close Intercept function was successful.
1	An incorrect host presentation space was specified.
2	A parameter error occurred.
9	A system error occurred.
10	The function is not supported by the emulation program.

Remarks

1. The returned event object or semaphore is in a non-signaled state when the start request function returns. The event object is in the signaled state each time a close request occurs. To receive notification of multiple close request events, put the event object into the signaled state each time using SetEvent or the [Query Close Intercept \(42\)](#) function.
2. After using this function, your application program can use the [Query Close Intercept \(42\)](#) function to determine when a close request has occurred. The application can wait on the returned event object to determine when the event has occurred.
3. This is not an exclusive call. Multiple applications can request this function for the same short session ID.
4. If there are no applications intercepting close requests for a session, any subsequent close requests selected by the user from the emulator operations dialog result in a normal stop requested for that session.

5.2.1.43 Start Communication Notification (80)

The Start Communication Notification function begins the process by which your EHLLAPI application can determine whether the specified session is connected to a host.

After using this function, the application can use [Query Communication Event \(81\)](#) to

determine whether the session is connected or disconnected.

3270	5250
Yes	Yes

Prerequisite Calls

There are no prerequisite calls for this function.

Call Parameters

	Enhanced Interface
Function Number	Must be 80
Data String	Preallocated structure; see the following table
Length	16
PSPosition	NA

The calling data structure contains these elements

Byte	Definition
1	A 1-character presentation space short name (PSID).
2-4	Reserved
5	One of the following values: The character C asks for notification when the session either disconnects or connects to the host. The character A requests the asynchronous mode of notification. When A is specified, position 9-12 returns the address of an event object (Windows). The character C must be placed in position 13. The character M requests the asynchronous message mode of the notification. When M is specified, the event selection character C must be placed in position 13.
6-8	Reserved
9-12	When M is specified in position 5, the window handle of the window that receives the message should be set. The message is a return value of RegisterWindowMessage (PCSHLL)--(not zero).
13	This should contain the character C if position 5 is A or M.
14-16	Reserved

Data String

If A (asynchronous mode) is specified in position 5 of the calling data structure and the function is completed successfully, the following data string is returned:

Byte	Definition
1	A 1-character presentation space short-name (PSID)
2-8	Reserved
9-12	4-byte binary value in which the event object handle is returned by EHLAPI. The application can wait for this event object.

If M (asynchronous message mode) is specified in position 5 of the calling data structure and the function is completed successfully, the following data string is returned:

Byte	Definition
1	A 1-character presentation space short-name (PSID)

2-8	Reserved
9-10	Task ID of asynchronous message mode

When the session connects or disconnects an application window receives a message. The message is the return value of RegisterWindow Message (PCSHLL). The wParam contains the Task ID returned by the function call. HIWORD of lParam contains a 21 if the session is connected to the host or a 22 if the session is disconnected. The LOWORD of lParam contains the function number 80.

Return Parameters

Return Code	Definition
0	The function was successful
1	An incorrect PSID was specified
2	An error was made in designating parameters
9	A system error was encountered

Remarks

1. An application program can issue this function for multiple host sessions. The [Query Communication Event \(81\)](#) function can be used to determine the session communication status.
2. If the application chooses the asynchronous option, it can use the Windows SDK call WaitForSingleObject to wait until the sessions communication status has changed.
3. The event object is initially in a non-signaled state. It is signaled each time an event occurs. To receive notification for multiple events the application must put the event object into the non-signaled state each time it is signaled, by using the Windows SDK call ResetEvent, or by using function 81 Query Communications Event.
4. Multiple calls to this function with the same options from the same application will be ignored.
5. This is not exclusive to one application. Several applications can request this function for the same Session ID.

5.2.1.44 Start Host Notification (23)

The Start Host Notification function begins the process by which your EHLLAPI application program determines if the host presentation space or OIA have been updated.

After using this function, your application program can use the [Query Host Update \(24\)](#) function to determine when a host event has occurred.

3270	5250
Yes	Yes

Prerequisite Calls

There are no prerequisite calls for this function.

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 23	
Data String	Preallocated string; see the following table	
Length	6 or 7 implied	16
PS Position	NA	

The calling data string contains these elements:

Byte		Definition
Standard	Enhanced	
1	1	One of the following values: A 1-character presentation space short name (PSID) A blank or null indicating a request for the host-connected host presentation space
	2-4	Reserved.
2	5	One of the following values: <ul style="list-style-type: none"> The character B asking for notification of both host presentation space and OIA updates. The character O asking for notification of only OIA updates. The character P asking for notification of only host presentation space updates. The character A requesting the asynchronous mode of the notification. When A is specified, position 9-12 returns the address of an event object. The event selection character B, O, or P must be placed in position 13. The character M requesting the asynchronous message mode of the notification. When M is specified, the event selection character B, O, or P must be placed in position 13 (7 for 16-bit). The character E asking for notification of completion during a printer session.
	6-8	Reserved.
3-4	9-12	When M is specified in position 5 (2 for 16-bit), the window handle of the window that receives the message should be set. The message is a return value of RegisterWindowMessage (PCSHLL) (not equal 0).
7	13	One of the following values if position 5 (2 for 16-bit) is A or M: <ul style="list-style-type: none"> The character B asking for notification of both host presentation space and OIA updates The character O asking for notification of only OIA updates The character P asking for notification of only host presentation update.
	14-16	Reserved.

Return Parameters

This function returns a data string and a return code.

Data String

If A (asynchronous mode of notification) is specified in position 5 and the function is completed successfully, the following data string is returned:

Byte		Definition
Standard	Enhanced	
1	1	A 1-character presentation space short name (PSID).

	2-8	Reserved.
	9-12	4-byte value in which the event object address is returned by EHLAPI. The application can wait for this event object (32-bit only).

Data String

If M (asynchronous message mode) is specified in position 5 (2 for standard interface) and the function is completed successfully, the following data string is returned:

Byte		Definition
Standard	Enhanced	
1	1	A 1-character presentation space short name (PSID)
	2-8	Reserved
3-4	9-10	Task ID of asynchronous message mode

Note:

If OIA or presentation space is updated, an application window receives a message. The message is a return value of RegisterWindowMessage (PCSHLL). The wParam parameter contains the Task ID returned by the function call. HIWORD of lParam contains Return Code 21 (shows the OIA is updated), 22 (shows the host presentation space is updated), or 23 (shows both the OIA and the host presentation space are updated), and LOWORD of lParam parameter contains function number 23.

Return Code

The following codes are defined:

Return Code	Definition
0	The Start Host Notification function was successful.
1	An incorrect host presentation space was specified.
2	An error was made in designating parameters.
9	A system error was encountered.

Remarks

1. An application program can issue this function for multiple host sessions. The [Pause \(18\)](#) function can notify the application when one or more host sessions (PS, OIA, or both of them) are updated. The [Query Host Update \(24\)](#) function can be used to determine whether a PS, OIA, or both of them have been updated.
2. If the application chooses the asynchronous option, it can wait for the returned event object or semaphore to determine when a host event has occurred.
3. The event object or semaphore is initially in a non-signaled state and is signaled each time an appropriate event occurs. To receive notification for multiple events, the application must put the event object into the non-signaled state each time it has been signaled using either the ResetEvent or the [Query Host Update \(24\)](#) function.
4. An application cannot request Start Host Notification more than once with the same options.
5. This is not an exclusive call. Multiple applications can request this function for the same short session ID.

5.2.1.45 Start Keystroke Intercept (50)

The Start Keystroke Intercept function allows a workstation application to filter any keystrokes sent to a session by a terminal operator. After a call to this function, keystrokes are intercepted and saved until the keystroke queue overflows or until the [Stop Keystroke Intercept \(53\)](#) function or [Reset System \(21\)](#) function is called. The intercepted keystrokes can be:

- Received through the [Get Key \(51\)](#) function and sent to the same or another session with the [Send Key \(3\)](#) function
- Accepted or rejected through the [Post Intercept Status \(52\)](#) function
- Replaced by other keystrokes with the [Send Key \(3\)](#) function
- Used to trigger other processes

3270	5250
Yes	Yes

Prerequisite Calls

There are no prerequisite calls for this function.

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 50	
Data String	See the following table	
Length	Keystroke buffer size EHLAPI allocates 32 bytes minimum for this buffer.	
PS Position	NA	

The calling data string contains:

Byte		Definition
Standard	Enhanced	
1	1	One of the following values: <ul style="list-style-type: none"> • A specific host presentation space short name (PSID) • A blank or null indicating a request for the host-connected host presentation space
	2-4	Reserved.
2	5	An option code character: <ul style="list-style-type: none"> • D for AID keystrokes only. • L for all keystrokes. • M for requesting the asynchronous message mode of the notification (Windows only). When M is specified, a code character D, or L must be placed in position 13 (7 for 16-bit).
	6-8	Reserved.
3-4	9-12	When M is specified in position 5 (2 for 16-bit), the window handle of the window that receives the message should be set. The message is a return value of RegisterWindowMessage (PCSHLL) (not equal 0).
7	13	One of the following values if position 5 (2 for 16-bit) is M: <ul style="list-style-type: none"> • D for AID keystrokes only. • L for all keystrokes.

	14-16	Reserved.
--	-------	-----------

Data String

If M (asynchronous message mode) is specified in position 5 (2 for standard interface) and the function is completed successfully, the following data string is returned:

Byte		Definition
Standard	Enhanced	
1	1	A 1-character presentation space short name (PSID)
	2-8	Reserved
3-4	9-10	Task ID of asynchronous message mode

Note:

If a user sends keystrokes to a session, an application window receives a message. The message is a return value of RegisterWindowMessge (PCSHLL). The wParam parameter contains the Task ID returned by the function call. HIWORD of lParam parameter contains return code 0, which shows that the function was successful, and LOWORD of lParam parameter contains function number 50.

Return Parameters

Return Code	Explanation
0	The Start Keystroke Intercept function was successful.
1	An incorrect presentation space was specified.
2	An incorrect option was specified.
4	The execution of the function was inhibited because the target presentation space was busy.
9	A system error was encountered. Release is being used.

Remarks

- If a return code of 31 occurs for the [Get Key \(51\)](#) function, either:
 - Increase the value of the calling length parameter for this function, or
 - Execute the [Get Key \(51\)](#) function more frequently.
 An intercepted keystroke occupies 3 bytes in the buffer. The next intercepted keystroke is placed in the adjacent 3 bytes. When the [Get Key \(51\)](#) function retrieves a keystroke (first-in first-out, or FIFO), the 3 bytes that it occupied are made available for another keystroke. By increasing the size of the buffer or the rate at which keystrokes are retrieved from the buffer, you can eliminate buffer overflow.
 In the PC/3270, another way to eliminate return code 31 is to operate the PC/3270 emulator in the resume mode.
- If option code D is provided, EHLLAPI writes intercepted non-AID keys to the presentation space to which they were originally intended, and returns only AID keys to the application.
- Call the [Stop Keystroke Intercept \(53\)](#) function before exiting your EHLLAPI application. Otherwise, keystroke interception remains enabled with unpredictable results.

5.2.1.46 Start Playing Macro (110)

The Start Playing Macro function invokes a macro. The macro will be executed in the connected session.

3270	5250
Yes	Yes

Prerequisite Calls

[Connect Presentation Space \(1\)](#)

Call Parameters

	Standard Interface	
Function Number	Must be 110	
Data String	See the following table	
Length	Length of macro name, plus 3	
PS Position	NA	

Byte		Definition
Standard	Enhanced	
1-2		Reserved
3-n		Null terminated macro name

Return Parameters

Return Code	Explanation
0	The Start Playing Macro function was successful.
1	The program is not connected to a host session.
2	An error was made in specifying parameters.
9	A system error was encountered.

5.2.1.47 Stop Close Intercept (43)

The Stop Close Intercept function allows the application to turn off the [Start Close Intercept \(41\)](#) function. After the application has issued the Stop Close Intercept function, subsequent close requests result in a normal stop sent to the logical terminal session.

3270	5250
Yes	Yes

Prerequisite Calls

[Start Close Intercept \(41\)](#)

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 43	
Data String	1-character short session ID of the host presentation space	
Length	1	Must be 4
PS Position	NA	

The calling data string can contain:

Byte		Definition
Standard	Enhanced	
1	1	A 1-character presentation space short name (PSID)
	2-4	Reserved

Return Parameters

Return Code	Explanation
0	The Stop Close Intercept function was successful.
1	An incorrect host presentation space was specified.
2	An error was made in specifying parameters.
8	No previous Start Close Intercept (41) function was issued.
9	A system error occurred.
12	The session stopped.

5.2.1.48 Stop Communication Notification (82)

The Stop Communication Notification function disables the capability of the [Query Communication Event \(81\)](#) function to determine whether any communication events have occurred in the specified Session.

3270	5250
Yes	Yes

Prerequisite Calls

[Start Communication Notification \(80\)](#)

Call Parameters

	Enhanced Interface
Function Number	Must be 82
Data String	1-character short name of the host presentation space, or a blank or null indicating request for updates to the host-connected

	presentation space
Length	4 is implied
PSPosition	NA

The calling data structure contains these elements:

Byte	Definition
1	A 1-character presentation space short name (PSID)
2-4	Reserved

Return Parameters

Return Code	Definition
0	The function was successful
1	An incorrect PSID was specified
8	No prior call to Start Communication Notification (80) function was called for the PSID
9	A system error was encountered

5.2.1.49 Stop Host Notification (25)

The Stop Host Notification function disables the capability of the [Query Host Update \(24\)](#) function to determine if the host presentation space or OIA has been updated. This function also stops host events from affecting the [Pause \(18\)](#) function.

3270	5250
Yes	Yes

Prerequisite Calls

[Start Host Notification \(23\)](#)

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 121	
Data String	See the following note	
Length	1 is implied	Must be 4
PS Position	NA	

The calling data string can contain:

Byte		Definition
Standard	Enhanced	
1	1	A 1-character presentation space short name (PSID)
	2-4	Reserved

Note:

1-character short name of the target presentation space ID, or a blank or a null to

indicate a request for the host-connected presentation space.

Return Parameters

Return Code	Definition
0	The Stop Host Notification function was successful.
1	An incorrect host presentation space was specified.
8	No previous Start Host Notification (23) function was issued.
9	A system error was encountered.

5.2.1.50 Stop Keystroke Intercept (53)

The Stop Keystroke Intercept function ends your application program's ability to intercept keystrokes.

3270	5250
Yes	Yes

Prerequisite Calls

[Start Keystroke Intercept \(50\)](#)

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 53	
Data String	Short name of the target presentation space (PSID)	
Length	1 is implied	Must be 4
PS Position	NA	

The calling data string can contain:

Byte		Definition
Standard	Enhanced	
1	1	A 1-character presentation space short name (PSID)
	2-4	Reserved

Return Parameters

Return Code	Explanation
0	The Stop Keystroke Intercept function was successful.
1	An incorrect presentation space was specified.
8	No prior Start Keystroke Intercept (50) function was called for this presentation space.
9	A system error was encountered.

5.2.1.51 Wait (4)

The Wait function checks the status of the host-connected presentation space. If the session is waiting for a host response (indicated by XCLOCK (X []) or XSYSTEM), the Wait function causes EHLLAPI to wait up to 1 minute to see if the condition clears.

3270	5250
Yes	Yes

Prerequisite Calls

[Connect Presentation Space \(1\)](#)

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 4	
Data String	NA	
Length	NA	
PS Position	NA	

Return Parameters

Return Code	Definition
0	The keyboard is unlocked and ready for input.
1	Your application program is not connected to a valid session.
4	Timeout while still in XCLOCK (X []) or XSYSTEM.
5	The keyboard is locked.
9	A system error was encountered.

Remarks

1. The Wait function is used to give host requests like those made by the [Send Key \(3\)](#) function the time required to be completed.
2. Using the [Set Session Parameters \(9\)](#) function, you can request the TWAIT, LWAIT, or the NWAIT option. You can use this function to see if the host OIA is inhibited.
3. The Wait function is satisfied by the host unlocking the keyboard. Therefore, a return code of 0 does not necessarily mean that the transaction has been completed. To verify completion of the transaction, you should use the [Search Field \(30\)](#) function or [Search Presentation Space \(6\)](#) function combined with the Wait function to look for expected keyword prompts.

5.2.1.52 Window Status (104)

The Window Status function allows the application to query or change a window's presentation space size, location, or visible state.

3270	5250
Yes	Yes

Prerequisite Calls

[Connect Window Services \(101\)](#)

Call Parameters

	Standard Interface	Enhanced Interface
Function Number	Must be 104	
Data String	See the following table	
Length	16 or 20	24 or 28
PS Position	NA	

The calling data string can contain:

Byte		Definition
Standard	Enhanced	
1	1	A 1-character presentation space short name (PSID)
	2-4	Reserved
2	5	A request option value, select one of the following values: <ul style="list-style-type: none"> • X'01' for set status Note: When the session is embedded In-Place in a compound OLE document, the set form of this function (byte 5 = X'01') always returns 0 but has no effect. <ul style="list-style-type: none"> • X'02' for query for status • X'03' for query for extended status
	6	Reserved

If the request option value is X'01' (set status):

Byte		Definition
Standard	Enhanced	
3-4	7-8	A 16- or 32-bit word containing the status set bits if the request option is 1 (set status). The following codes are valid return values if the request option is set status: <ul style="list-style-type: none"> • X'0001' Change the window size. (Not valid with minimize, maximize, restore, or move.) • X'0002' Move the window. (Not valid with minimize, maximize, size, or restore.) X'0004' ZORDER window replacement. • X'0008' Set the window to visible. • X'0010' Set the window to invisible. • X'0080' Activate the window. (Sets focus to window and places it in the foreground unless ZORDER is specified. In this case, the ZORDER placement is used.) • X'0100' Deactivate the window. (Deactivates the window and makes the window the bottom window unless ZORDER is also specified. In this case, the ZORDER placement is used.) • X'0400' Set the window to minimized. (Not valid with maximize, restore, size, or move.) • X'0800' Set the window to maximized. (Not valid with minimize, restore, size, or move.) • X'1000' Restore the window. (Not valid with minimize, maximize, size, or

Byte	Standard	Enhanced	Definition
			move.)
5-6	9-12		A 16- or 32-bit word containing the X window position coordinate. (Ignored if the move option is not set.)
7-8	13-16		A 16- or 32-bit word containing the Y window position coordinate. (Ignored if the move option is not set.)
9-10	17-20		A 16- or 32-bit word containing the X window size in device units. (Ignored if the size option is not set.)
11-12	21-24		A 16- or 32-bit word containing the Y window size in device units. (Ignored if the size option is not set.)
13-16	25-28		A 16- or 32-bit word containing a window handle for relative window placement. These two words are only for the set option. (Ignored if the ZORDER option is not set.) Valid values are as follows: <ul style="list-style-type: none"> • X'00000003' Place in front of all sibling windows. • X'00000004' Place behind all sibling windows.

If the request option value is X'02' (query for status):

Byte	Standard	Enhanced	Definition
3-4	7-8		A 16- or 32-bit word containing X'0000' if the request option is 2 (query for status). The following codes are possible return values if the request option is query for status. More than one state is possible. <ul style="list-style-type: none"> • X'0008' The window is visible. • X'0010' The window is invisible. • X'0080' The window is activated. • X'0100' The window is deactivated. • X'0400' The window is minimized. • X'0800' The window is maximized.
5-6	9-12		A 16- or 32-bit word containing the X window position coordinate. (Ignored if the move option is not set.)
7-8	13-16		A 16- or 32-bit word containing the Y window position coordinate. (Ignored if the move option is not set.)
9-10	17-20		A 16- or 32-bit word containing the X window size in device units. (Ignored if the size option is not set.)
11-12	21-24		A 16- or 32-bit word containing the Y window size in device units. (Ignored if the size option is not set.)
13-16	25-28		A 16- or 32-bit word containing a window handle for relative window placement. These two words are only for the set option. (Ignored if the ZORDER option is not set.) Valid values are as follows: <ul style="list-style-type: none"> • X'00000003' Place in front of all sibling windows. • X'00000004' Place behind all sibling windows.

If the request option value is X'03' (query for extended status):

Byte	Standard	Enhanced	Definition
3-4	7-8		A 16- or 32-bit word containing X'0000' if the request option is 3 (query for extended status). The following codes are possible return values if the request option is query for extended status. More than one state is possible. <ul style="list-style-type: none"> • X'0008' The window is visible. • X'0010' The window is invisible. • X'0080' The window is activated. • X'0100' The window is deactivated. • X'0400' The window is minimized. • X'0800' The window is maximized.
5-6	9-10		A 16- or 32-bit word containing the current font size in the X-dimension. The value is in screen pels.
7-8	11-12		A 16- or 32-bit word containing the current font size in the Y-dimension. The value is in screen pels.

9-12	13-16	Reserved. This value is always zero.
13-14	17-18	A 16- or 32-bit word containing the row number of the first visible character of the presentation space. This value is usually one, unless the Fixed Size font option is in effect, and the window has been resized such that some of the presentation space is hidden.
15-16	19-20	A 16- or 32-bit word containing the column number of the first visible character of the presentation space.
17-20	21-24	A 16- or 32-bit word containing the presentation space window handle of the session.

Return Parameters

Return Code	Explanation
0	The Window Status function was successful.
1	The presentation space was not valid or not connected.
2	An incorrect option was specified.
9	A system error occurred.
12	The session stopped.

Remarks

The logical terminal (LT) windows use character cells. When resizing the LT windows, the LT rounds the number to prevent character cell truncation. The requested size and position might be slightly different from what was requested. Follow the set option with a query option to determine the final Presentation Manager window position and size. All x and y coordinate positions and sizes are in pels.

5.2.2 WinHLLAPI Extension Functions

This chapter describes the extension functions provided when using WinHLLAPI programming support.

- [Asynchronous Functions](#)
- [Initialization and Termination Functions](#)
- [Blocking Routines](#)

5.2.2.1 Asynchronous Functions

The following sections describe the WinHLLAPI asynchronous functions.

- [WinHLLAPIAsync](#)
- [WinHLLAPICancelAsyncRequest](#)

5.2.2.1.1 WinHLLAPIAsync

This entry point is used for six WinHLLAPI functions that often take a long time to complete. With WinHLLAPIAsync, the function will be launched asynchronously and will not interfere with the continued progression of the calling application.

These functions are:

- [Receive File \(91\)](#)
- [Start Close Intercept \(41\)](#)
- [Send File \(90\)](#)
- [Start Host Notify \(23\)](#)
- [Start Keystroke Intercept \(50\)](#)
- [Wait \(04\)](#)

Syntax

`HANDLE WinHLLAPIAsync (HWND hWnd, LPWORD lpnFunction, LPBYTE lpData, LPWORD lpnLength, LPWORD lpnRetC)*`

The parameter list is the same as WinHLLAPI except a window handle is required before the function number. Since the function operates asynchronously, its completion is signaled by a registered message. The window handle is required as the target of the message.

There are two messages that must be registered by the WinHLLAPI application through calls to RegisterWindowsMessage() with the strings WinHLLAPIAsync (for all functions except 90 and 91) and WinHLLAPIAsyncFileTransfer (for functions 90 and 91). The standard format is as follows:

WPARAM

Contains the Task Handle returned by the original function call.

LPARAM

The high word contains the error code and the low word contains the original function number.

5.2.2.1.1.1 Receive File (91)

This function transfers a file from the PC to the Host.

Prerequisite Functions

There are no prerequisite functions for this function.

Syntax

`WinHLLAPIAsync (hWnd, lpwFunction, lpbyString, lpwLength, lpwReturnCode)`

Call Parameters

Parameter	Description
Data String	RECEIVE command parameters.
Data Length	Length of Data String. NA if session option EOT is specified.

PS Position	NA
-------------	----

Return Codes

Code	Description
WHLLOK	File transfer started successfully.
WHLLPARAMETERERROR	Parameter error or Data Length is zero or greater than 255.
WHLLFTXCOMPLETE	File transfer complete.
WHLLFTXSEGMENTED	Transfer is complete with segmented records.
WHLLSYSERROR	The function failed due to a system error.
WHLLTRANSABORTED	File transfer aborted, either due to the user clicking the cancel button or because the timeout period has elapsed.
WHLLFILENOTFOUND	PC file not found.
WHLLFTXCOMPLETECICS	File transfer was successful (transfer to CICS).
WHLLACCESSDENIED	Access denied to PC file.
WHLLMEMORY	Insufficient memory.
WHLLINVALIDENVIRONMENT	Invalid environment.

Remarks

Only one file transfer operation is supported per connected Host session.

The function initiates the file transfer and immediately returns control to your Windows HLLAPI application. This frees your application to perform other tasks while the file transfer is occurring. Once initiated the function will regularly post WinHLLAPIAsyncFileTransfer messages to the window specified by hWnd. These messages will notify the WinHLLAPI application of the status of the transfer and send a final message when the transfer is complete.

wParm

Is the status indicator: the high byte contains the Session ID, the low byte contains the status. If the low byte is zero, the file transfer is still in progress. If the low byte is one, the file transfer has completed.

lParm

If the low byte of wParm is zero (in progress), lParm is the number of bytes transferred. If the low byte wParm is one (completed), lParm is the completion code.

5.2.2.1.1.2 Start Close Intercept (41)

This function intercepts user requests to close Personal Communications.

Prerequisite Functions

There are no prerequisite functions for this function.

Syntax

WinHLLAPIAsync (hWnd, lpwFunction, lpbyString, lpwLength, lpwReturnCode)

Call Parameters

Parameter	Description
Data String	A 5-byte string for returned semaphore address. The first byte is the session short name of the session to query, or space or null for the current session.
Data Length	Must be specified.
PS Position	NA

Return Parameters

Parameter	Description
Data String	A 5-byte string with the following format: Byte 1 Session short name, or space or null for the current session Bytes 2-5 Semaphore address.

Return Code

Code	Description
WHLLOK	The function was successful.
WHLLNOTCONNECTED	An invalid presentation space was specified.
WHLLPARAMETERERROR	An invalid option was specified.
WHLLSYSERROR	The function failed due to a system error.
WHLLCANCEL	The asynchronous function was cancelled.

Remarks

Once enabled, Host notification remains enabled until you call [Stop Close Intercept \(43\)](#) or [WinHLLAPICancelAsyncRequest\(\)](#). Initially, the semaphore is set. After using this function, close requests from the user are discarded and the semaphore is cleared.

The function initiates close intercept and immediately returns control to your Windows HLLAPI application. This frees your application to perform other tasks while waiting for close requests. When a close request occurs, the function will notify the window specified by hWnd with the registered message [WinHLLAPIAsync](#).

5.2.2.1.1.3 Send File (90)

This function transfers a file from the PC to the Host.

Prerequisite Functions

There are no prerequisite functions for this function.

Syntax

WinHLLAPIAsync (hWnd, lpwFunction, lpbyString, lpwLength, lpwReturnCode)

Call Parameters

Parameter	Description
Data String	SEND command parameters.
Data Length	Length of Data String. NA if session option EOT is specified.
PS Position	NA

Return Codes

Code	Description
WHLLOK	File transfer started successfully.
WHLLPARAMETERERROR	Parameter error or Data Length is zero or greater than 255.
WHLLFTXCOMPLETE	File transfer complete.
WHLLFTXSEGMENTED	Transfer is complete with segmented records.
WHLLSYSERROR	The function failed due to a system error.
WHLLTRANSABORTED	File transfer aborted, either due to the user clicking the cancel button or because the timeout period has elapsed.
WHLLFILENOTFOUND	PC file not found.
WHLLFTXCOMPLETECICS	File transfer was successful (transfer to CICS).
WHLLACCESSDENIED	Access denied to PC file.
WHLLMEMORY	Insufficient memory.
WHLLINVALIDENVIRONMENT	Invalid environment.

Remarks

Only one file transfer operation is supported per connected Host session.

The function initiates the file transfer and immediately returns control to your Windows HLLAPI application. This frees your application to perform other tasks while the file transfer is occurring. Once initiated the function will regularly post WinHLLAPIAsyncFileTransfer messages to the window specified by hWnd. These messages will notify the WinHLLAPI application of the status of the transfer and send a final message when the transfer is complete.

wParm

Is the status indicator: the high byte contains the Session ID, the low byte contains the status. If the low byte is zero, the file transfer is still in progress. If the low byte is one, the file transfer has completed.

lParm

If the low byte of wParm is zero (in progress), lParm is the number of bytes transferred. If the low byte wParm is one (completed), lParm is the completion code.

5.2.2.1.1.4 Start Host Notification (23)

This function enables you to notify your WinHLLAPI application of changes in the Host Session Presentation Space (PS) or Operation Information Area (OIA).

Prerequisite Functions

There are no prerequisite functions for this function.

Syntax

`WinHLLAPIAsync (hWnd, lpwFunction, lpbyString, lpwLength, lpwReturnCode)`

Call Parameters

Parameter	Description
Data String	A 7-byte string in the following format: <ul style="list-style-type: none"> • Byte 1 Short name session ID of the desired Host session, or space or null for the current Host session. • Byte 2 Notification mode. "P" for presentation space update only, "O" for OIA update only, "B" for both presentation space and OIA updates. When calling WinHLLAPIAsync, this position can be "A". • Byte 3-6 Not used. Provided for compatibility with older applications. • Byte 7 Reserved or replaced with one of the following if using WinHLLAPIAsync and A in byte 2: P for presentation space update only, O for OIA update only; and B for both presentation space and OIA updates.
Data Length	Length of Host event buffer (256 recommended).
PS Position	NA

Return Parameters

Parameter	Description
Data String	Same as Data String on the call.

Return Codes

Code	Description
WHLLOK	Host notification enabled.
WHLLNOTCONNECTED	The specified Host session is invalid.
WHLLPARAMETERERROR	One of more parameters are invalid.
WHLLSYSERROR	The function failed due to a system error.
WHLLCANCEL	The asynchronous function was cancelled.

Remarks

Once enabled, Host notification is enabled until you call [Stop Host Notification \(25\)](#) or [WinHLLAPICancelAsyncRequest\(\)](#). The function initiates host notification and immediately returns control to your Windows HLLAPI application. This frees your application to perform other tasks while waiting for host updates. When an update occurs, the function will notify the window specified by hWnd with the registered

message WinHLLAPIAsync.

5.2.2.1.1.5 Start Keystroke Intercept (50)

This function intercepts keystrokes sent to a session by the user.

Prerequisite Functions

There are no prerequisite functions for this function.

Syntax

[WinHLLAPIAsync](#) (hWnd, lpwFunction, lpbyString, lpwLength, lpwReturnCode)

Call Parameters

Parameter	Description
Data String	A 6-byte string in the following format: <ul style="list-style-type: none"> • Byte 1 Session short name, or space or null for the current Host session. • Byte 2 Keystroke intercept code. "D" causes only AID keystrokes to be intercepted; "L" causes all keystrokes to be intercepted. • Bytes 3-6 Reserved
Data Length	Variable (256 is recommended)
PS Position	NA

Return Code

Code	Description
WHLLOK	Keystroke intercept has been initiated.
WHLLNOTCONNECTED	The Host session presentation space is invalid.
WHLLPARAMETERERROR	One or more parameters are invalid.
WHLLPSBUSY	Session is busy.
WHLLSYSERROR	Function failed due to a system error.
WHLLCANCEL	Asynchronous function was cancelled.

Remarks

The function initiates keystroke intercept and immediately returns control to your Windows HLLAPI application. This frees your application to perform other tasks while waiting for keystrokes. Once initiated, the function will post a [WinHLLAPIAsync](#) message to the window specified by hWnd whenever the user sends a key to the PS. After notification, the intercepted keystrokes can be handled in any way that is allowed by a normal EHLLAPI application. Take note that the keystroke buffer is of limited size so each keystroke should be handled and removed from the buffer.

5.2.2.1.1.6 Wait (4)

This function determines whether the Host session is in an inhibited state. If, for some reason, the session is in an inhibited state, this function will signal your application with a message when either the inhibited state expires or your wait period has expired.

The amount of time to wait is set with the [Set Session Parameters \(9\)](#) function.

Prerequisite Functions

[Connect Presentation Space \(1\)](#)

Syntax

```
WinHLLAPIAsync(hWnd, lpwFunction, lpbyString, lpwLength, lpwReturnCode)
```

Call Parameters

Parameter	Description
Data String	NA
Data Length	NA
PS Position	NA

Return Codes

Code	Description
WHLLOK	The PS is uninhibited and ready for input.
WHLLNOTCONNECTED	Your WinHLLAPI application is not connected to a valid host session.
WHLLPSBUSY	Function timed out while still inhibited.
WHLLNHIBITED	The PS is inhibited.
SHLLSYSERROR	The function failed due to a system error.
WHLLCANCEL	The asynchronous function was cancelled.

Remarks

Asynchronous Wait is used to notify the calling application when the inhibited state of the PS is expired. When inhibited state has expired, this version of Wait will post a [WinHLLAPIAsync](#) message to the window specified by the hWnd. The session options TWAIT, LWAIT, and NWAIT affect the length of time that this function will wait. See [Set Session Parameters \(9\)](#) for details on these session options.

Note:

If NWAIT is specified in the session parameters and the application registers using revision 1.1 of the WinHLLAPI implementation, the [WINHLLAPIAsync](#) call will work the same as the WinHLLAPI call and not send a message. If revision 1.0 is being used then Wait will return a message immediately with the inhibited status of the PS.

5.2.2.1.2 WinHLLAPICancelAsyncRequest

This function cancels an outstanding asynchronous function launched by a call to WinHLLAPIAsync().

Syntax

`int WinHLLAPICancelAsyncRequest (HANDLE hAsyncTask, WORD wFunction)`

Parameters

hAsyncTask

The handle returned by `WinHLLAPIAsync()` when the function was initiated.

wFunction

The function number of the asynchronous task to cancel. Because this parameter is required for revision 1.1 but not in 1.0, it is optional.

With this function, any asynchronous task previously initiated by a call to [WinHLLAPIAsync\(\)](#) may be canceled while still outstanding.

Returns

The return value indicates if the specified function was, in fact, canceled. If the function was canceled then the return value is `WHLLOK (0)`. If the outstanding asynchronous function was not cancelled, one of the following codes will be returned.

WHELLINVALID

hAsyncTask is not a valid task handle.

WHELLALREADY

The asynchronous task specified by *hAsyncTask* has already completed.

5.2.2.2 Initialization and Termination Functions

The following section describes the initialization and termination functions of WinHLLAPI programming support.

- [WinHLLAPI Startup](#)
- [WinHLLAPI Cleanup](#)

5.2.2.2.1 WinHLLAPI Startup

This function is used to register the application with the WinHLLAPI implementation and should be called before any other call to the WinHLLAPI implementation. This implementation supports Versions 1.0 and 1.1 of the WinHLLAPI specification. The WinHLLAPI application should negotiate version compatibility with this function.

Syntax

`int WinHLLAPIStartup(WORD wVersionRequired, LPWHELLAPIDATA lpData)`

Parameters

wVersionRequired

This is the version required by the WinHLLAPI application. The low byte contains the major version number and the high byte contains the minor version (or revision) number.

lpData

This is a pointer to a WHLLAPIDATA structure which will receive the implementations version number and a string describing the WinHLLAPI implementation provider. The WHLLAPIDATA structure is defined as:

```
#define WHLLDESCRIPTION_LEN 127
typedef struct tagWHLLAPIDATA
{
    WORD wVersion;
    Char szDescription[WHLLDESCRIPTION_LEN + 1];
}WHLLAPIDATA, * PWHLLAPIDATA, FAR *LPWHLLAPIDATA;
```

Returns

The return value indicates success or failure of registering the WinHLLAPI application with the implementation. If registration was successful, the return value is WHLLOK (zero). Otherwise, it is one of the following:

WHLLSYSNOTREADY

Indicates that the underlying network subsystem is unavailable.

WHLLVERNOTSUPPORTED

Indicates that the version requested is not provided by this implementation. This implementation supports Versions 1.0 and 1.1 only.

5.2.2.2 WinHLLAPI Cleanup

The WinHLLAPI specification recommends that this function be used by the WinHLLAPI application to de-register from the WinHLLAPI implementation.

Syntax

```
BOOL WinHLLAPICleanup()
```

Returns

Returns TRUE if the unregistration was successful. Otherwise, it returns FALSE.

5.2.2.3 Blocking Routines

The following sections describe the blocking routines supported by WinHLLAPI programming.

- [WinHLLAPIIsBlocking](#)
- [WinHLLAPISetBlockingHook](#)
- [WinHLLAPIUnhookBlockingHook](#)
- [WinHLLAPICancelBlockingCall](#)

Note:

Although blocking routines are supported for WinHLLAPI compliance, use of them is not recommended. Use of the [WinHLLAPIAsync](#) functions are the recommended method for asynchronous processing.

5.2.2.3.1 WinHLLAPIIsBlocking

This function tells the calling WinHLLAPI application thread whether it is in the process of executing a blocking call. A blocking call is any synchronous function that takes a long time to execute and does not return until complete. There are five blocking calls in this implementation of WinHLLAPI. The blocking calls are: [Get Key \(51\)](#), [Wait \(4\)](#), [Pause \(18\)](#), [Send File \(90\)](#), and [Receive File \(91\)](#).

Syntax

```
BOOL WinHLLAPIIsBlocking()
```

Returns

If the WinHLLAPI application thread is in the middle of a blocking call, the function returns TRUE, otherwise, it returns FALSE.

Remarks

Because the default blocking-hook allows messages to be processed during blocking calls, it is possible to call the blocking call again.

5.2.2.3.2 WinHLLAPISetBlockingHook

This function sets an application-defined procedure to be executed while waiting for the completion of a blocking call. A blocking call is any synchronous function that takes a long time to execute and does not return until complete. There are five blocking calls in this implementation of WinHLLAPI. The blocking calls are: [Get Key \(51\)](#), [Wait \(4\)](#), [Pause \(18\)](#), [Send File \(90\)](#), and [Receive File \(91\)](#).

Syntax

```
FARPROC WinHLLAPISetBlockingHook(FARPROC lpfnBlockingHook)
```

Parameters

lpfnBlockingHook

This is a pointer to the new blocking procedure.

Description

The WinHLLAPI implementation has a default blocking procedure that consists of nothing more than a message handler. This default mechanism is shown in the following example:

```
BOOL DefaultBlockingHook
{
    MSG msg;

    if (PeekMessage (&msg, NULL, 0, 0, xFPM_NOREMOVE))
    {
        if(msg.message == WM_QUIT)
        {
            return FALSE;
        }
        PeekMessage (&msg, NULL, 0, 0, PM_REMOVE);
        TranslateMessage (&msg);
        DispatchMessage (&msg);
    }
    return TRUE;
}
```

The blocking hook is implemented on a per-thread basis. A blocking hook set by this function will stay in effect for the thread until it is replaced by another call to [WinHLLAPISetBlockingHook\(\)](#) or until the default is restored by a call to [WinHLLAPIUnhookBlockingHook\(\)](#).

The Blocking function must return FALSE if it receives a WM_QUIT message so WinHLLAPI can return control to the application to process the message and terminate gracefully. Otherwise, the function should return TRUE.

Returns

This function returns a pointer to the blocking function being replaced.

5.2.2.3.3 WinHLLAPIUnhookBlockingHook

This function restores the default blocking-hook for the calling thread.

Syntax

```
BOOL WinHLLAPIUnhookBlockingHook()
```

Returns

This function returns TRUE if the default blocking mechanism was successfully

restored, otherwise it returns FALSE.

5.2.2.3.4 WinHLLAPICancelBlockingCall

This function cancels an executing blocking call in the current thread. A blocking call is any synchronous function that takes a long time to execute and does not return until complete. There are five blocking calls in this implementation of WinHLLAPI. The blocking calls are [Get Key \(51\)](#), [Wait \(4\)](#), [Pause \(18\)](#), [Send File \(90\)](#), and [Receive File \(91\)](#). If one of these is blocking calls are cancelled, the cancelled function will return WHLLCANCEL.

Syntax

```
int WinHLLAPICancelBlockingCall()
```

Returns

The return value indicates if the specified function was, in fact, canceled. If the function was canceled, then the return value is WHLLOK (0). If there are no outstanding blocking functions, then the following return code will be returned:

WHLLINVALID

Indicates that there is no blocking call currently executing.

5.3 OHIO Reference

The file zohio.ocx has to be registered and placed in the directory as zhllap32.dll, which works in out-of-process mode, allowing any z/Scope version to be controlled by OHIO by a separate process. The entry points are taken from this file. The class ohiomanager is accessed through zohio.ohiomanager. For example:

Visual Basic:

```
CreateObject("zOHIO.OHIOManager")
```

Javascript:

```
var OHIOManager = new  
ActiveXObject("zOHIO.OHIOManager");
```

For a complete reference of the OHIO API, refer to the following sections:

- [OHIO Containment Hierarchy](#)
- [OHIO Inheritance Hierarchy](#)
- [Programming Reference](#)

5.3.1 Examples

For some examples, please refer to the following topics:

- [Visual Basic](#)

- [Html](#)
- [Delphi](#)

5.3.1.1 Visual Basic

```
Public Class Form1
    Dim AxOHIOManager1 As zOHIO.OHIOManager

    Private Sub RefreshSessionsCombo()
        Dim i As Integer

        ComboBox1.Items.Clear()
        For i = 0 To (AxOHIOManager1.Sessions.Count - 1)
            ComboBox1.Items.Add(AxOHIOManager1.Sessions.Item(i).sessionName)
        Next i
        ComboBox1.SelectedIndex = 0
    End Sub

    Private Function IfThen(ByVal BoolValue As Boolean, ByVal OnTrue As String, ByVal
OnFalse As String) As String
        If BoolValue Then
            Return OnTrue
        Else
            Return OnFalse
        End If
    End Function

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles Button1.Click
        Try
            AxOHIOManager1.Sessions.Refresh()
            RefreshSessionsCombo()
        Catch ex As Exception
            MsgBox(ex.ToString)
        Exit Sub
        End Try
        'TextBox3.Text = AxOHIOManager1.Sessions.Item(TextBox1.Text).Screen.Fields.
Item(Val(TextBox2.Text)).String
    End Sub

    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles Button4.Click
        Try
            TextBox3.Text = AxOHIOManager1.Sessions.Item(ComboBox1.Text).Screen.
Fields.Item(Val(MaskedTextBox1.Text)).String
        Catch ex As Exception
            MsgBox(ex.ToString)
        End Try
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles Button2.Click
        Try
            AxOHIOManager1.Sessions.Item(ComboBox1.Text).Screen.Fields.Item(Val
```

```

(MaskedTextBox1.Text)).String = TextBox3.Text
    Catch ex As Exception
        MsgBox(ex.ToString)
    End Try
End Sub

Private Sub Button7_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles Button7.Click
    Dim SessionType As String
    Dim FieldIdx As Integer

    ListBox1.Items.Clear()
    ListBox1.Items.Add("Session Name: " + AxOHIOManager1.Sessions.Item
(ComboBox1.Text).sessionName)
    If (AxOHIOManager1.Sessions.Item(ComboBox1.Text).SessionType = zOHIO.
OHIO_TYPE.OHIO_TYPE_3270) Then
        SessionType = "3270"
    ElseIf (AxOHIOManager1.Sessions.Item(ComboBox1.Text).SessionType = zOHIO.
OHIO_TYPE.OHIO_TYPE_5250) Then
        SessionType = "5250"
    ElseIf (AxOHIOManager1.Sessions.Item(ComboBox1.Text).SessionType = zOHIO.
OHIO_TYPE.OHIO_TYPE_VT) Then
        SessionType = "VT"
    Else
        SessionType = "UNKNOWN"
    End If
    ListBox1.Items.Add("Session Type: " + SessionType)
    ListBox1.Items.Add("Rows: " + Str(AxOHIOManager1.Sessions.Item(ComboBox1.
Text).Screen.Rows))
    ListBox1.Items.Add("Cols: " + Str(AxOHIOManager1.Sessions.Item(ComboBox1.
Text).Screen.Columns))
    ListBox1.Items.Add("Fields: " + Str(AxOHIOManager1.Sessions.Item(ComboBox1.
Text).Screen.Fields.Count))
    With AxOHIOManager1.Sessions.Item(ComboBox1.Text)
        For FieldIdx = 0 To (.Screen.Fields.Count - 1)
            ListBox1.Items.Add(" Field N°: " + Str(FieldIdx) + Chr(13))
            ListBox1.Items.Add(" Start Pos: " + Str(.Screen.Fields.Item(FieldIdx).
Start) + Chr(13))
            ListBox1.Items.Add(" End Pos: " + Str(.Screen.Fields.Item(FieldIdx).
End) + Chr(13))
            ListBox1.Items.Add(" Length: " + Str(.Screen.Fields.Item(FieldIdx).
Length) + Chr(13))
            ListBox1.Items.Add(" Data: " + .Screen.Fields.Item(FieldIdx).String +
Chr(13))
            ListBox1.Items.Add(" Attributes: " + Chr(13))
            ListBox1.Items.Add(" Modified: " + IfThen(.Screen.Fields.Item
(FieldIdx).Modified, "Yes", "No") + Chr(13))
            ListBox1.Items.Add(" Protected: " + IfThen(.Screen.Fields.Item
(FieldIdx).Protected, "Yes", "No") + Chr(13))
            ListBox1.Items.Add(" Numeric: " + IfThen(.Screen.Fields.Item
(FieldIdx).Numeric, "Yes", "No") + Chr(13))
            ListBox1.Items.Add(" HighIntensity: " + IfThen(.Screen.Fields.Item
(FieldIdx).HighIntensity, "Yes", "No") + Chr(13))

```

```
        ListBox1.Items.Add("        PenSelectable: " + IfThen(.Screen.Fields.Item
(FieldIdx).PenSelectable, "Yes", "No") + Chr(13))
        ListBox1.Items.Add("        Hidden: " + IfThen(.Screen.Fields.Item
(FieldIdx).Hidden, "Yes", "No") + Chr(13))
        ListBox1.Items.Add
("-----" + Chr
(13))
    Next FieldIdx
End With
End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles Button3.Click
    Try
        AxOHIOManager1.Sessions.Item(ComboBox1.Text).Screen.SetString
(TextBox3.Text, Val(MaskedTextBox2.Text))
    Catch ex As Exception
        MsgBox(ex.ToString)
    End Try
End Sub

Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles Button5.Click
    Try
        AxOHIOManager1.Sessions.Item(ComboBox1.Text).Screen.SendKeys
(TextBox3.Text, Val(MaskedTextBox2.Text))
    Catch ex As Exception
        MsgBox(ex.ToString)
    End Try
End Sub

Private Sub Button8_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles Button8.Click
    If Not AxOHIOManager1.Sessions.Item(ComboBox1.Text).Screen.
WaitForNewScreen(10000) Then
        MsgBox("TIMEOUT: Wait for New Screen")
    End If
End Sub

Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles Button6.Click
    Dim AidKeyCode As Integer

    If (ComboBox2.Text = "ATTN") Then
        AidKeyCode = zOHIO.OHIO_AID_KEY.OHIO_AID_KEY_ATTN
    ElseIf (ComboBox2.Text = "BTAB") Then
        AidKeyCode = zOHIO.OHIO_AID_KEY.OHIO_AID_KEY_BACKTAB
    ElseIf (ComboBox2.Text = "CLEAR") Then
        AidKeyCode = zOHIO.OHIO_AID_KEY.OHIO_AID_KEY_CLEAR
    ElseIf (ComboBox2.Text = "ENTER") Then
        AidKeyCode = zOHIO.OHIO_AID_KEY.OHIO_AID_KEY_ENTER
    ElseIf (ComboBox2.Text = "FTAB") Then
        AidKeyCode = zOHIO.OHIO_AID_KEY.OHIO_AID_KEY_TABFORWARD
```

```
ElseIf (ComboBox2.Text = "HELP") Then
    AidKeyCode = zOHIO.OHIO_AID_KEY.OHIO_AID_KEY_HELP
ElseIf (ComboBox2.Text = "PA1") Then
    AidKeyCode = zOHIO.OHIO_AID_KEY.OHIO_AID_KEY_PA1
ElseIf (ComboBox2.Text = "PA2") Then
    AidKeyCode = zOHIO.OHIO_AID_KEY.OHIO_AID_KEY_PA2
ElseIf (ComboBox2.Text = "PA3") Then
    AidKeyCode = zOHIO.OHIO_AID_KEY.OHIO_AID_KEY_PA3
ElseIf (ComboBox2.Text = "PF1") Then
    AidKeyCode = zOHIO.OHIO_AID_KEY.OHIO_AID_KEY_PF1
ElseIf (ComboBox2.Text = "PF10") Then
    AidKeyCode = zOHIO.OHIO_AID_KEY.OHIO_AID_KEY_PF10
ElseIf (ComboBox2.Text = "PF11") Then
    AidKeyCode = zOHIO.OHIO_AID_KEY.OHIO_AID_KEY_PF11
ElseIf (ComboBox2.Text = "PF12") Then
    AidKeyCode = zOHIO.OHIO_AID_KEY.OHIO_AID_KEY_PF12
ElseIf (ComboBox2.Text = "PF13") Then
    AidKeyCode = zOHIO.OHIO_AID_KEY.OHIO_AID_KEY_PF13
ElseIf (ComboBox2.Text = "PF14") Then
    AidKeyCode = zOHIO.OHIO_AID_KEY.OHIO_AID_KEY_PF14
ElseIf (ComboBox2.Text = "PF15") Then
    AidKeyCode = zOHIO.OHIO_AID_KEY.OHIO_AID_KEY_PF15
ElseIf (ComboBox2.Text = "PF16") Then
    AidKeyCode = zOHIO.OHIO_AID_KEY.OHIO_AID_KEY_PF16
ElseIf (ComboBox2.Text = "PF17") Then
    AidKeyCode = zOHIO.OHIO_AID_KEY.OHIO_AID_KEY_PF17
ElseIf (ComboBox2.Text = "PF18") Then
    AidKeyCode = zOHIO.OHIO_AID_KEY.OHIO_AID_KEY_PF18
ElseIf (ComboBox2.Text = "PF19") Then
    AidKeyCode = zOHIO.OHIO_AID_KEY.OHIO_AID_KEY_PF19
ElseIf (ComboBox2.Text = "PF2") Then
    AidKeyCode = zOHIO.OHIO_AID_KEY.OHIO_AID_KEY_PF2
ElseIf (ComboBox2.Text = "PF20") Then
    AidKeyCode = zOHIO.OHIO_AID_KEY.OHIO_AID_KEY_PF20
ElseIf (ComboBox2.Text = "PF21") Then
    AidKeyCode = zOHIO.OHIO_AID_KEY.OHIO_AID_KEY_PF21
ElseIf (ComboBox2.Text = "PF22") Then
    AidKeyCode = zOHIO.OHIO_AID_KEY.OHIO_AID_KEY_PF22
ElseIf (ComboBox2.Text = "PF23") Then
    AidKeyCode = zOHIO.OHIO_AID_KEY.OHIO_AID_KEY_PF23
ElseIf (ComboBox2.Text = "PF24") Then
    AidKeyCode = zOHIO.OHIO_AID_KEY.OHIO_AID_KEY_PF24
ElseIf (ComboBox2.Text = "PF3") Then
    AidKeyCode = zOHIO.OHIO_AID_KEY.OHIO_AID_KEY_PF3
ElseIf (ComboBox2.Text = "PF4") Then
    AidKeyCode = zOHIO.OHIO_AID_KEY.OHIO_AID_KEY_PF4
ElseIf (ComboBox2.Text = "PF5") Then
    AidKeyCode = zOHIO.OHIO_AID_KEY.OHIO_AID_KEY_PF5
ElseIf (ComboBox2.Text = "PF6") Then
    AidKeyCode = zOHIO.OHIO_AID_KEY.OHIO_AID_KEY_PF6
ElseIf (ComboBox2.Text = "PF7") Then
    AidKeyCode = zOHIO.OHIO_AID_KEY.OHIO_AID_KEY_PF7
ElseIf (ComboBox2.Text = "PF8") Then
```

```
        AidKeyCode = zOHIO.OHIO_AID_KEY.OHIO_AID_KEY_PF8
    ElseIf (ComboBox2.Text = "PF9") Then
        AidKeyCode = zOHIO.OHIO_AID_KEY.OHIO_AID_KEY_PF9
    ElseIf (ComboBox2.Text = "PGDOWN") Then
        AidKeyCode = zOHIO.OHIO_AID_KEY.OHIO_AID_KEY_PGDOWN
    ElseIf (ComboBox2.Text = "PGUP") Then
        AidKeyCode = zOHIO.OHIO_AID_KEY.OHIO_AID_KEY_PGUP
    ElseIf (ComboBox2.Text = "PRINT") Then
        AidKeyCode = zOHIO.OHIO_AID_KEY.OHIO_AID_KEY_PRINT
    ElseIf (ComboBox2.Text = "SYSREQ") Then
        AidKeyCode = zOHIO.OHIO_AID_KEY.OHIO_AID_KEY_SYSREQ
    Else
        Exit Sub
    End If
Try
    AxOHIOManager1.Sessions.Item(ComboBox1.Text).Screen.SendAid
(AidKeyCode)
Catch ex As Exception
    MsgBox(ex.ToString)
End Try
End Sub

Private Sub Button9_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles Button9.Click
    Button9.Enabled = False
Try
    AxOHIOManager1 = New zOHIO.OHIOManager
    AxOHIOManager1.Sessions.AddSession("", "A")
    'AxOHIOManager1.Sessions.AddSession("", "B")
    ListBox1.Enabled = True
    TextBox3.Enabled = True
    MaskedTextBox1.Enabled = True
    MaskedTextBox2.Enabled = True
    ComboBox1.Enabled = True
    ComboBox2.Enabled = True
    ComboBox2.SelectedIndex = 3
    ListBox1.Items.Clear()
    TextBox3.Clear()
    MaskedTextBox1.Text = "0"
    MaskedTextBox2.Text = "0"
    Button1.Enabled = True
    Button2.Enabled = True
    Button3.Enabled = True
    Button4.Enabled = True
    Button5.Enabled = True
    Button6.Enabled = True
    Button7.Enabled = True
    Button8.Enabled = True
    Button10.Enabled = True
Catch ex As Exception
    Button1.Enabled = False
    Button2.Enabled = False
    Button3.Enabled = False
```

```
        Button4.Enabled = False
        Button5.Enabled = False
        Button6.Enabled = False
        Button7.Enabled = False
        Button8.Enabled = False
        Button9.Enabled = True
        Button10.Enabled = False
        ListBox1.Enabled = False
        TextBox3.Enabled = False
        MaskedTextBox1.Enabled = False
        MaskedTextBox2.Enabled = False
        ComboBox1.Enabled = False
        ComboBox2.Enabled = False
        MsgBox(ex.ToString)
        AxOHIOManager1 = Nothing
    Exit Sub
End Try
RefreshSessionsCombo()
End Sub

Private Sub Button10_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles Button10.Click
    Button1.Enabled = False
    Button2.Enabled = False
    Button3.Enabled = False
    Button4.Enabled = False
    Button5.Enabled = False
    Button6.Enabled = False
    Button7.Enabled = False
    Button8.Enabled = False
    Button9.Enabled = True
    Button10.Enabled = False
    ListBox1.Enabled = False
    TextBox3.Enabled = False
    MaskedTextBox1.Enabled = False
    MaskedTextBox2.Enabled = False
    ComboBox1.Enabled = False
    ComboBox2.Enabled = False
    AxOHIOManager1 = Nothing
End Sub
End Class
```

5.3.1.2 Html

```
<html>

<head>
<title>z/Scope Web-to-Host</title>
<script >
var OhioManager = new ActiveXObject("zOHIO.OHIOManager");
var OhioSession;
function Connect()
{
```

```

        OhioSession = OhioManager.OpenSession("", "*");
        if (OhioSession==null) {
            alert("no session");
        };
        Refresh();
    }
function SendKeys(keys)
{
    if (OhioSession==null) {
        alert("no session");
    };
    if (keys==null) {
        keys = document.all("keys").value;
    } else {
        keys = document.all("keys").value+keys;
    };
    OhioSession.Screen.SendKeys(keys,OhioSession.Screen.Cursor);
    OhioSession.Screen.WaitForUnlock();
    Refresh();
}

function Refresh()
{
    if (OhioSession==null) {
        alert("no session");
    };
    document.all("screen").innerHTML = "";
    for (var i=1;i<OhioSession.Screen.Rows;i++) {
        var line = OhioSession.Screen.GetData((i-1)*OhioSession.Screen.
Columns+1,i*OhioSession.Screen.Columns, 1);
        document.all("screen").innerHTML = document.all("screen").innerHTML +
"<pre>" +line + "</pre>";
    }
}
</script>
</head>

<body bgcolor="#ffffff" text="#000000" style="MARGIN: 0px" scroll="no"
onLoad="Connect()">
<pre id="screen"></pre>
<form>
<input type="TEXT" name="keys" id="keys" style="WIDTH: 299px; HEIGHT: 22px"
size="38">
<input type="button" onClick="Refresh()" value="Refresh">
<input type="button" onClick="SendKeys()" value="SendKeys">
<input type="button" onClick="SendKeys('@E')" value="ENTER">
</form>
</body>
</html>

```

5.3.1.3 Delphi

```
unit TestX;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, zOHIO_TLB, StdCtrls, StrUtils, ComCtrls,
  Mask, ExtCtrls, uTnbHllOhioConsts;

type
  TForm1 = class(TForm)
    Memo1 : TMemo;
    Button1 : TButton;
    Button2 : TButton;
    Button3 : TButton;
    Label1: TLabel;
    Label2: TLabel;
    Edit2: TEdit;
    Button4: TButton;
    Button5: TButton;
    MaskEdit1: TMaskEdit;
    Button6: TButton;
    Bevel1: TBevel;
    Label3: TLabel;
    MaskEdit2: TMaskEdit;
    Bevel3: TBevel;
    Label4: TLabel;
    ComboBox1: TComboBox;
    Button7: TButton;
    Button8: TButton;
    Button9: TButton;
    Label5: TLabel;
    ComboBox2: TComboBox;
    Bevel2: TBevel;
    Button10: TButton;
    procedure FormDestroy(Sender : TObject);
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure Button6Click(Sender: TObject);
    procedure Button5Click(Sender: TObject);
    procedure Button7Click(Sender: TObject);
    procedure Button8Click(Sender: TObject);
    procedure Button9Click(Sender: TObject);
    procedure Button10Click(Sender: TObject);
  private
    { Private declarations }
    fOhioManager : IOHIOManager;
    procedure RefreshSessionsCombo;
    procedure ScreenChangeEvent(Sender : TObject; const session: IOHIOSession);
```

```
    procedure SessionStateEvent(Sender : TObject; const session: IOHIOSession);
    procedure SessionLockEvent(Sender : TObject; const session: IOHIOSession);
    procedure SessionUnlockEvent(Sender : TObject; const session: IOHIOSession);
public
    { Public declarations }
end;

var
    Form1 : TForm1;

implementation

{$R *.dfm}

procedure TForm1.Button10Click(Sender: TObject);
begin
    if not fOhioManager.Sessions.Item[ComboBox2.Text].Screen.WaitForNewScreen
(10000) then
        ShowMessage('TIMEOUT: Wait for New Screen');
end;

procedure TForm1.Button1Click(Sender: TObject);
var OhioMgr : TOHIOManager;
begin
    Button1.Enabled := FALSE;
    ComboBox1.ItemIndex := 0;
    try
        OhioMgr := TOHIOManager.Create(Self);
        OhioMgr.OpenSession('', 'A');
        OhioMgr.OpenSession('', 'B');
        OhioMgr.OnScreenChange := ScreenChangeEvent;
        OhioMgr.OnSessionState := SessionStateEvent;
        OhioMgr.OnSystemLock := SessionLockEvent;
        OhioMgr.OnSystemUnLock := SessionUnlockEvent;
        FOhioManager := OhioMgr.ControlInterface as IOHIOManager;
        Button2.Enabled := TRUE;
        Button3.Enabled := TRUE;
        Button4.Enabled := TRUE;
        Button5.Enabled := TRUE;
        Button6.Enabled := TRUE;
        Button7.Enabled := TRUE;
        Button8.Enabled := TRUE;
        Button9.Enabled := TRUE;
        Button10.Enabled := TRUE;
        Memo1.Enabled := TRUE;
        Edit2.Enabled := TRUE;
        MaskEdit1.Enabled := TRUE;
        MaskEdit2.Enabled := TRUE;
        ComboBox1.Enabled := TRUE;
        ComboBox2.Enabled := TRUE;
        Memo1.Clear;
        Edit2.Clear;
        MaskEdit1.Text := '0';
```

```
MaskEdit2.Text := '0';
except
on E : Exception do
begin
Memo1.Lines.Add('ERROR: ' + E.Message);
Button1.Enabled := TRUE;
Button2.Enabled := FALSE;
Button3.Enabled := FALSE;
Button4.Enabled := FALSE;
Button5.Enabled := FALSE;
Button6.Enabled := FALSE;
Button7.Enabled := FALSE;
Button8.Enabled := FALSE;
Button9.Enabled := FALSE;
Button10.Enabled := FALSE;
Memo1.Enabled := FALSE;
Edit2.Enabled := FALSE;
MaskEdit1.Enabled := FALSE;
MaskEdit2.Enabled := FALSE;
ComboBox1.Enabled := FALSE;
ComboBox2.Enabled := FALSE;
if Assigned(fOhioManager) then
FreeAndNil(fOhioManager);
Exit;
end;
end;
RefreshSessionsCombo;
end;

procedure TForm1.Button2Click(Sender: TObject);
var
FieldIdx : integer;
SessionType : string;
begin
Memo1.Clear;
Memo1.Lines.Add('Session Name: ' + fOhioManager.Sessions.Item[ComboBox2.Text].
SessionName);
if (fOhioManager.Sessions.Item[ComboBox2.Text].SessionType = OHIO_TYPE_3270)
then
SessionType := '3270'
else if (fOhioManager.Sessions.Item[ComboBox2.Text].SessionType =
OHIO_TYPE_5250) then
SessionType := '5250'
else
SessionType := 'VT';
Memo1.Lines.Add('Session Type: ' + SessionType);
Memo1.Lines.Add('Rows: ' + IntToStr(fOhioManager.Sessions.Item[ComboBox2.Text].
Screen.Rows));
Memo1.Lines.Add('Cols: ' + IntToStr(fOhioManager.Sessions.Item[ComboBox2.Text].
Screen.Columns));
Memo1.Lines.Add('Fields: ' + IntToStr(fOhioManager.Sessions.Item[ComboBox2.Text].
Screen.Fields.Count));
with fOhioManager.Sessions.Item[ComboBox2.Text] do
```

```

begin
  for FieldIdx := 0 to (Screen.Fields.Count - 1) do
    begin
      Memo1.Lines.Add(' Field N°: ' + IntToStr(FieldIdx));
      Memo1.Lines.Add(' Start Row: ' + IntToStr(Screen.Fields.Item[FieldIdx].
StartObj.Row));
      Memo1.Lines.Add(' Start Col: ' + IntToStr(Screen.Fields.Item[FieldIdx].
StartObj.Col));
      Memo1.Lines.Add(' End Row: ' + IntToStr(Screen.Fields.Item[FieldIdx].EndObj.
Row));
      Memo1.Lines.Add(' End Col: ' + IntToStr(Screen.Fields.Item[FieldIdx].EndObj.
Col));
      Memo1.Lines.Add(' Length: ' + IntToStr(Screen.Fields.Item[FieldIdx].Length));
      Memo1.Lines.Add(' Data: ' + Screen.Fields.Item[FieldIdx].String_);
      Memo1.Lines.Add(' Attributes: ');
      Memo1.Lines.Add(' Modified: ' + IfThen(Screen.Fields.Item[FieldIdx].
Modified, 'Yes', 'No'));
      Memo1.Lines.Add(' Protected: ' + IfThen(Screen.Fields.Item[FieldIdx].
Protected_, 'Yes', 'No'));
      Memo1.Lines.Add(' Numeric: ' + IfThen(Screen.Fields.Item[FieldIdx].
Numeric, 'Yes', 'No'));
      Memo1.Lines.Add(' HighIntensity: ' + IfThen(Screen.Fields.Item[FieldIdx].
HighIntensity, 'Yes', 'No'));
      Memo1.Lines.Add(' PenSelectable: ' + IfThen(Screen.Fields.Item[FieldIdx].
PenSelectable, 'Yes', 'No'));
      Memo1.Lines.Add(' Hidden: ' + IfThen(Screen.Fields.Item[FieldIdx].
Hidden, 'Yes', 'No'));
      Memo1.Lines.Add
('-----');
    end;
    Application.ProcessMessages;
  end;
end;

procedure TForm1.Button3Click(Sender: TObject);
begin
  fOhioManager.Sessions.Item[ComboBox2.Text].Screen.SendKeys(Edit2.Text, StrToInt
(MaskEdit2.Text));
end;

procedure TForm1.Button4Click(Sender: TObject);
begin
  try
    fOhioManager.Sessions.Item[ComboBox2.Text].Screen.Fields.Item[StrToInt
(MaskEdit1.Text)].String_ := Edit2.Text;
  except
    on E: Exception do
      ShowMessage(E.Message);
    end;
  end;
end;

procedure TForm1.Button5Click(Sender: TObject);
begin

```

```
try
  fOhioManager.Sessions.Item[ComboBox2.Text].Screen.SetString(Edit2.Text,
StrToInt(MaskEdit2.Text));
except
  on E: Exception do
    ShowMessage(E.Message);
end;
end;

procedure TForm1.Button6Click(Sender: TObject);
begin
  try
    Edit2.Text := fOhioManager.Sessions.Item[ComboBox2.Text].Screen.Fields.Item
[StrToInt(MaskEdit1.Text)].String_;
  except
    on E: Exception do
      ShowMessage(E.Message);
    end;
  end;
end;

procedure TForm1.Button7Click(Sender: TObject);
begin
  try
    fOhioManager.Sessions.Item[ComboBox2.Text].Screen.SendAid
(ConvertStrAidToIntAid(ComboBox1.Text));
  except
    on E: Exception do
      ShowMessage(E.Message);
    end;
  end;
end;

procedure TForm1.Button8Click(Sender: TObject);
begin
  Button1.Enabled := TRUE;
  Button2.Enabled := FALSE;
  Button3.Enabled := FALSE;
  Button4.Enabled := FALSE;
  Button5.Enabled := FALSE;
  Button6.Enabled := FALSE;
  Button7.Enabled := FALSE;
  Button8.Enabled := FALSE;
  Button8.Enabled := FALSE;
  Button9.Enabled := FALSE;
  Button10.Enabled := FALSE;
  Memo1.Enabled := FALSE;
  Edit2.Enabled := FALSE;
  MaskEdit1.Enabled := FALSE;
  MaskEdit2.Enabled := FALSE;
  ComboBox1.Enabled := FALSE;
  ComboBox2.Enabled := FALSE;
  Memo1.Clear;
  Edit2.Clear;
  MaskEdit1.Text := '0';
```

```
MaskEdit2.Text := '0';
ComboBox2.Clear;
if Assigned(fOhioManager) then
  fOhioManager := nil;
end;

procedure TForm1.Button9Click(Sender: TObject);
begin
  if Assigned(fOhioManager) then
  begin
    try
      fOhioManager.Sessions.Refresh;
      RefreshSessionsCombo;
    except
      Button8Click(Button8);
    end;
  end;
end;

procedure TForm1.FormDestroy(Sender : TObject);
begin
  Button8Click(Button8);
end;

procedure TForm1.RefreshSessionsCombo;
var
  i : integer;
begin
  ComboBox2.Clear;
  for i := 0 to (fOhioManager.Sessions.Count - 1) do
    ComboBox2.Items.Add(fOhioManager.Sessions.Item[i].SessionName);
  ComboBox2.ItemIndex := 0;
end;

procedure TForm1.ScreenChangeEvent(Sender: TObject; const session: IOHIOSession);
begin
  Memo1.Lines.Add
  ('-----');
  Memo1.Lines.Add('| Session "' + session.SessionName + '" Screen has changed |');
  Memo1.Lines.Add
  ('-----');
end;

procedure TForm1.SessionStateEvent(Sender: TObject; const session: IOHIOSession);
begin
  Memo1.Lines.Add
  ('-----');
  Memo1.Lines.Add('| Session "' + session.SessionName + '" State has changed |');
  Memo1.Lines.Add
  ('-----');
end;
```

```

procedure TForm1.SessionLockEvent(Sender: TObject; const session: IOHIOSession);
begin
  Memo1.Lines.Add('-----');
  Memo1.Lines.Add('| Session "' + session.SessionName + '" Lock |');
  Memo1.Lines.Add('-----');
end;

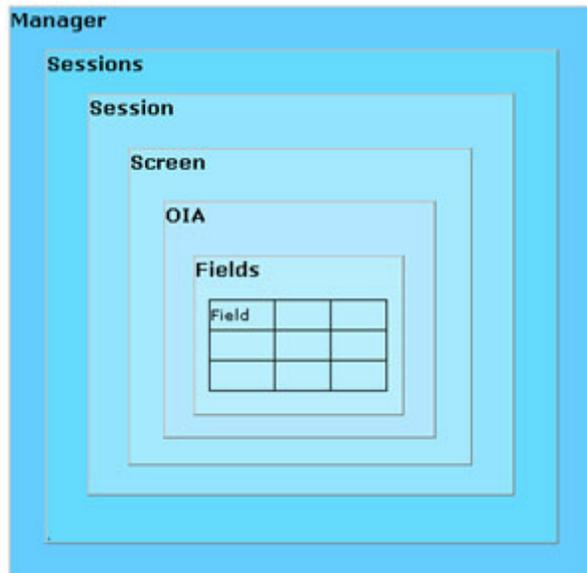
procedure TForm1.SessionUnlockEvent(Sender: TObject; const session:
IOHIOSession);
begin
  Memo1.Lines.Add('-----');
  Memo1.Lines.Add('| Session "' + session.SessionName + '" Unlock |');
  Memo1.Lines.Add('-----');
end;

end.

```

5.3.2 OHIO Containment Hierarchy

The following is the Ohio containment hierarchy:



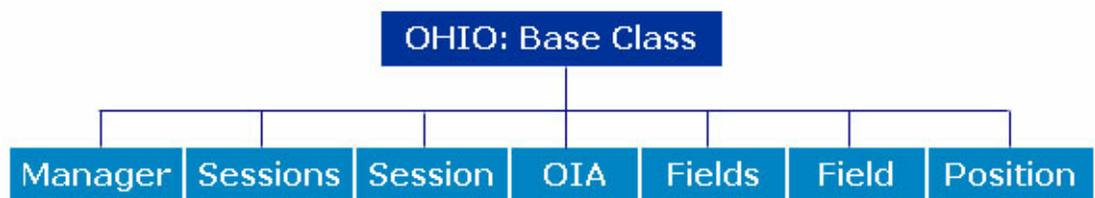
Additional utility classes:

- OhioPosition

Note: "1" based counting is used throughout this document for both positions (the first position on the screen is position 1, not position 0) and sizes (the first item in a collection is item 1, not item 0).

5.3.3 OHIO Inheritance Hierarchy

The Ohio inheritance hierarchy is:



5.3.4 Programming Reference

In this chapter we describe the following topics:

- Ohio Base Class
- [Class Definition](#)
- [Constants](#)

5.3.4.1 Class Definition

- [OhioManager](#)
- [OhioSessions](#)
- [OhioSession](#)
- [OhioScreen](#)
- [OhioFields](#)
- [OhioField](#)
- [OhioOIA](#)
- [OhioPosition](#)

5.3.4.1.1 OhioField

A field is the fundamental element of a virtual screen. A field includes both data and attributes describing the field. The OhioField class encapsulates a virtual screen field and provides methods for accessing and manipulating field attributes and data.

OhioField objects can be accessed only through the OhioFields object.

See Also

[Properties](#) and [Methods](#).

5.3.4.1.1.1 Methods

- [GetData](#)

Returns a character array containing the data from the Text, Color, Field or Extended plane of the virtual screen.

Visual Basic Syntax

```
[Array =] Field.GetData(plane As Integer)
```

Delphi Syntax

```
[TCharArr :=] Field.GetData(plane: integer);
```

Parameters

Parameter	Description
<i>plane</i>	A valid OHIO_PLANE value.

5.3.4.1.1.2 Properties

- [Attribute](#)
- [EndPos](#)
- [Hidden](#)
- [HighIntensity](#)
- [Length](#)
- [Modified](#)
- [Numeric](#)
- [PenSelectable](#)
- [ProtectedField](#)
- [StartPos](#)
- [String/Text](#)

Indicates the attribute byte for the field.

Visual Basic Syntax

```
[Integer =] Field.Attribute
```

Delphi Syntax

```
[Integer :=] Field.Attribute;
```

Returns the ending position of the field. The position can range from 1 to the size of the virtual screen. The ending position of a field is the position of the last character in the field.

Visual Basic Syntax

```
[TTnbOhioPosition =] Field.EndPos
```

Delphi Syntax

[TTnbOhioPosition :=] Field.EndPos;

Remarks

Returns an integer indicating the position of the last character in the field.

Indicates whether or not the field is hidden.

Visual Basic Syntax

[Boolean =] Field.Hidden

Delphi Syntax

[Boolean :=] Field.Hidden;

Remarks

This property returns True if the field is hidden, otherwise false.

Indicates whether or not the field is high-intensity.

Visual Basic Syntax

[Boolean =] Field.HighIntensity

Delphi Syntax

[Boolean :=] Field.HighIntensity;

Remarks

Returns True if the field is high intensity, otherwise False.

Returns the length of the field.

Visual Basic Syntax

[Integer =] Field.Length

Delphi Syntax

[Integer :=] Field.Length;

Remarks

A field's length can range from 1 to the size of the virtual screen.

Indicates whether or not the field has been modified.

Visual Basic Syntax

[Boolean =] Field.Modified

Delphi Syntax

[Boolean :=] Field.Modified;

Remarks

This property returns True if the field has been modified, otherwise False.

Indicates whether the field which contains the cursor is a numeric-only field.

Visual Basic Syntax

[Boolean =] Field.Numeric

Delphi Syntax

[Boolean :=] Field.Numeric;

Remarks

This property returns True if the cursor is in a numeric-only field, false otherwise.

Indicates whether or not the field is pen-selectable.

Visual Basic Syntax

[Boolean =] Field.PenSelectable

Delphi Syntax

[Boolean :=] Field.PenSelectable;

Remarks

This property returns True if the field is pen-selectable, otherwise False.

Indicates whether or not the field is protected.

Visual Basic Syntax

```
[Boolean =] Field.Protected
```

Delphi Syntax

```
[Boolean :=] Field.Protected;
```

Remarks

This property returns True if the field is protected, otherwise False.

Indicates the starting position of the field.

Visual Basic Syntax

```
[TTnbOhioPosition =] Field.StartPos
```

Delphi Syntax

```
[TTnbOhioPosition :=] Field.StartPos;
```

Remarks

The position can range from 1 to the size of the virtual screen. The starting position of a field is the position of the first character in the field.

Returns the entire text plane of the virtual screen as a string. String and Text are both the same properties, but String can't be used in Delphi as a function name.

Visual Basic Syntax

```
Field.String [= String]
```

Delphi Syntax

```
Field.Text; [:= String]
```

Remarks

All null characters and Field Attribute characters are returned as blank space characters.

5.3.4.1.2 OhioFields

OhioFields contains a collection of the fields in the virtual screen. It provides methods to iterate through the fields, find fields based on location, and find fields containing a given string. Each element of the collection is an instance of OhioField.

OhioFields can only be accessed through OhioScreen using the Fields property. OhioFields is a static view of the virtual screen and does not reflect changes made to the virtual screen after its construction. The field list can be updated with a new view of the virtual screen using the Refresh() method.

Note: All OhioFields objects returned by methods in this class are invalidated when Refresh() is called.

See Also

OhioFields [Properties](#) and [Methods](#).

5.3.4.1.2.1 Methods

- [Item](#)
- [FindByString](#)
- [FindByPosition](#)
- [Refresh](#)

Searches the collection for the target position and returns the OhioField object containing that position. If not found, returns a null.

Visual Basic Syntax

```
[TTnbOhioField =] Fields.FindByPosition(targetPosition As TTnbOhioPosition)
```

Delphi Syntax

```
[TTnbOhioField :=] Fields.FindByPosition(targetString: TTnbOhioPosition);
```

Parameters

Parameters	Description
<i>TargetPosition</i>	The target row and column.

Searches the collection for the target string and returns the OhioField object containing that string. The string must be totally contained within the field to be considered a match. If the target string is not found, a null will be returned.

Visual Basic Syntax

```
[TTnbOhioField =] Fields.FindByString(targetString As String, startPos As TTnbOhioPosition, length As Integer, dir As Integer, ignoreCase As Boolean)
```

Delphi Syntax

```
[TTnbOhioField :=] Fields.FindByString(targetString:string; startPos: TTnbOhioPosition; length:integer; dir:integer; ignoreCase:boolean);
```

Parameters

Parameter	Description
<i>targetString</i>	The target string.
<i>startPos</i>	The row and column where to start. The position is inclusive (for example, row 1, col 1 means that position 1,1 will be used as the starting location and 1,1 will be included in the search).
<i>length</i>	The length from startPos to include in the search.
<i>dir</i>	An OHIO_DIRECTION value.
<i>ignoreCase</i>	Indicates whether the search is case sensitive. True means that case will be ignored. False means the search will be case sensitive.

Indicates the OhioSession object at the given index. "One based" indexing is used in all Ohio collections. For example, the first OhioSession in this collection is at index 1.

Visual Basic Syntax

```
[TTnbOhioField =] Fields.Item(index As Integer)
```

Delphi Syntax

```
[TTnbOhioField :=] Fields.Item(index:integer);
```

Parameters

Parameters	Description
<i>index</i>	The index of the target OhioSession.

Returns null if no object with that name exists in this collection.

Updates the collection of OhioSession objects. All OhioSession objects that are available on the system at the time of the refresh will be added to the collection.

Indexing of OhioSession objects will not be preserved across refreshes.

Visual Basic Syntax

Fields.Refresh

Delphi Syntax

Fields.Refresh;

Parameters

None.

5.3.4.1.2.2 Properties

- [Columns](#)
- [Count](#)
- [Rows](#)

Indicates the number of columns in the presentation space.

Visual Basic Syntax

[Integer =] Fields.Columns

Delphi Syntax

[Integer :=] Fields.Columns;

Remarks

This property returns an integer representing the number of columns on the current host screen.

Returns the number of OhioSession objects contained in this collection.

Visual Basic Syntax

[Integer =] Fields.Count

Delphi Syntax

[Integer :=] Fields.Count;

Indicates the number of rows in the presentation space.

Visual Basic Syntax

```
[Integer =] Fields.Rows
```

Delphi Syntax

```
[Integer :=] Fields.Rows;
```

Remarks

This property returns an integer representing the number of rows on the current host screen.

5.3.4.1.3 OhioManager

The OhioManager is the central repository for access to all OHIO sessions. It contains a list of all OhioSession objects available on this system.

See Also

OhioManager [Properties](#) and [Methods](#).

5.3.4.1.3.1 Methods

- [OpenSession](#)
- [CloseSession](#)

Closes an OhioSession object. The OhioSession is considered invalid and is removed from the list of OhioSession objects. You can use this methods with the following syntaxes:

Visual Basic Syntax

```
Manager.CloseSession(SessionObject As TTnbOhioSession)
```

```
Manager.CloseSession(SessionName As String)
```

Delphi Syntax

```
Manager.CloseSession(SessionObject: TTnbOhioSession);
```

```
Manager.CloseSession(SessionName: string);
```

Parameters

Parameters	Description
<i>SessionObject</i>	The OhioSession to close.
<i>SessionName</i>	The SessionName of the OhioSession to close.

Returns an OhioSession object based on the search parameters provided.

Visual Basic Syntax

```
[TTnbOhioSession =] Manager.OpenSession(ConfigurationResource As Integer,
SessionName As String)
```

```
[TTnbOhioSession =] Manager.OpenSession(ConfigurationResource As
TTnbComApi, SessionName As String)
```

```
[TTnbOhioSession =] Manager.OpenSession(ConfigurationResource As String,
SessionName As String)
```

Delphi Syntax

```
[TTnbOhioSession :=] Manager.OpenSession(ConfigurationResource: integer;
SessionName: string);
```

```
[TTnbOhioSession :=] Manager.OpenSession(ConfigurationResource: TTnbComApi;
SessionName: string);
```

```
[TTnbOhioSession :=] Manager.OpenSession(ConfigurationResource, SessionName:
string);
```

Parameters

Parameter	Description
<i>ConfigurationResource</i>	A vendor specific string used to provide configuration information
<i>SessionName</i>	The unique name associated with an OhioSession

The parameters are used as follows:

- Is ConfigurationResource provided?
 - Yes - Is SessionName provided?
 - Yes - Is OhioSession object with matching SessionName available on the system?
 - Yes - Error, attempting to create an OhioSession object with a non-unique SessionName.
 - No - Create an OhioSession object using SessionName and ConfigurationResource.
 - No - Start a new OhioSession using ConfigurationResource and generating a new SessionName.

- No - Is SessionName provided?
- Yes - Is OhioSession object with matching SessionName available on the system?
- Yes - Return identified OhioSession object.
- No - Return null.
- No - Return null.

5.3.4.1.3.2 Properties

- [Sessions](#)

Returns an OhioSessions object containing the OhioSession objects available on this system. This list of objects is a static snapshot at the time the OhioSessions object is created.

Visual Basic Syntax

```
[OhioSessions =] Manager.Sessions
```

Delphi Syntax

```
[OhioSessions :=] Manager.Sessions;
```

5.3.4.1.4 OhioOIA

The operator information area of a host session. This area is used to provide status information regarding the state of the host session and location of the cursor.

An OhioOIA object can be obtained using the OIA() method on an instance of OhioScreen.

5.3.4.1.4.1 Properties

- [Alphanumeric](#)
- [CommCheckCode](#)
- [InputInhibited](#)
- [MachineCheckCode](#)
- [Numeric](#)
- [Owner](#)
- [ProgCheckCode](#)

Indicates whether the field which contains the cursor is an alphanumeric field. True if the cursor is in an alphanumeric field, false otherwise.

Visual Basic Syntax

```
[Boolean=] OIA.Alphanumeric
```

Delphi Syntax

[Boolean :=] OIA.Alphanumeric;

Remarks

This property returns True if the cursor is in an alphanumeric field, False in the other case.

Returns the communication check code.

Visual Basic Syntax

[Integer =] OIA.CommCheckCode

Delphi Syntax

[Integer :=] OIA.CommCheckCode;

Remarks

If InputInhibited returns OHIO_INPUTINHIBITED_COMMCHECK, this property will return the communication check code.

Indicates whether or not input is inhibited.

Visual Basic Syntax

[Integer =] OIA.InputInhibited

Delphi Syntax

[Integer :=] OIA.InputInhibited;

Remarks

If input is inhibited, SendKeys or SendAID calls to the OhioScreen are not allowed. Why input is inhibited can be determined from the value returned. If input is inhibited for more than one reason, the highest value is returned.

Specifies the machine check code.

Visual Basic Syntax

[Integer =] OIA.MachineCheckCode

Delphi Syntax

```
[Integer :=] OIA.MachineCheckCode;
```

Remarks

If InputInhibited returns OHIO_INPUTINHIBITED_MACHINECHECK, this property will return the machine check code.

Indicates whether the field which contains the cursor is a numeric-only field.

Visual Basic Syntax

```
[Boolean =] OIA.Numeric
```

Delphi Syntax

```
[Boolean :=] OIA.Numeric;
```

Remarks

This property returns True if the cursor is in a numeric-only field, false otherwise.

Indicates the owner of the host connection.

Visual Basic Syntax

```
[Integer =] OIA.Owner
```

Delphi Syntax

```
[Integer :=] OIA.Owner;
```

Remarks

The Integer returned by this property specifies one of the [OHIO_OWNER](#) values.

Returns the program check code.

Visual Basic Syntax

```
[Integer =] OIA.ProgCheckCode
```

Delphi Syntax

```
[Integer :=] OIA.ProgCheckCode;
```

Remarks

If `InputInhibited` returns `OHIO_INPUTINHIBITED_PROGCHECK`, this property will return the program check code.

5.3.4.1.5 OhioPosition

Holds row and column coordinates. An `OhioPosition` can be constructed by using [CreateOhioPosition\(\)](#) on any `Ohio` class.

See Also

`OhioPosition` [Properties](#) and [Methods](#).

5.3.4.1.5.1 Methods

- [Refresh](#)

Updates the collection of `OhioSession` objects. All `OhioSession` objects that are available on the system at the time of the refresh will be added to the collection.

Indexing of `OhioSession` objects will not be preserved across refreshes.

Visual Basic Syntax

```
Position.Refresh(TnbFields As TTnbFields, fromPos As Integer)
```

Delphi Syntax

```
Position.Refresh(TnbFields: TTnbFields; fromPos: integer);
```

Parameters

Parameters	Description
<i>TnbFields</i>	The fields to refresh.
<i>fromPos</i>	The position where to start.

5.3.4.1.5.2 Properties

- [Col](#)
- [Pos](#)
- [Row](#)

Indicates the column coordinate.

Visual Basic Syntax

Position.Col [= Integer]

Delphi Syntax

Position.Col; [: = Integer]

Gets or sets the current position.

Visual Basic Syntax

Position.Pos [= Integer]

Delphi Syntax

Position.Pos; [: = Integer]

Remarks

Value indicates the position to set to the Position object.

Indicates the row coordinate.

Visual Basic Syntax

Position.Row [= Integer]

Delphi Syntax

Position.Row; [: = Integer]

5.3.4.1.6 OhioScreen

OhioScreen encapsulates the host presentation space. The presentation space is a virtual screen which contains all the characters and attributes that would be seen on a traditional emulator screen. This virtual screen is the primary object for text-based interactions with the host. The OhioScreen provides methods that manipulate text, search the screen, send keystrokes to the host, and work with the cursor.

An OhioScreen object can be obtained from the Screen property of an instance of OhioSession.

The raw presentation space data is maintained in a series of planes which can be accessed by various methods within this class.

The text plane contains the actual characters in the presentation space. Most of

the methods in OhioScreen class work exclusively with the text plane.

The remaining planes contain the corresponding attributes for each character in the text plane. The color plane contains color characteristics. The field plane contains the field attributes.

The extended plane contains the extended field attributes. The color, field, and extended planes are not interpreted by any of the methods in this class.

See Also

OhioScreen [Properties](#), [Methods](#) and [Events](#).

5.3.4.1.6.1 Events

- [OnConnect](#)
- [OnConnectionFail](#)
- [OnDisconnect](#)
- [OnScreenChange](#)
- [OnSendAid](#)
- [OnSessionState](#)

Occurs after a successfully connection to the server is established.

Occurs after the connection to the server fails.

Occurs after a disconnection of the server.

This event is generated whenever the virtual screen is modified.

Occurs before an Aid key is to be sent.

Remarks

This event is fired when a [SendAid](#) or [SendKeys](#) methods are used to send an Aid Key and data to the mainframe. Occurs before the data is sent, allowing to take actions like filling edit fields or saving data typed in the emulator control.

Occurs when a session-state change takes place.

Remarks

A session state can be ssNoSession, ssSSCPLU and ssLULU. Any transition between the states fires this event.

5.3.4.1.6.2 Methods

- [GetData](#)
- [FindString](#)
- [Refresh](#)
- [SendAid](#)
- [SendKeys](#)
- [SetString](#)
- [Wait](#)
- [WaitFor](#)
- [WaitForNewScreen](#)
- [WaitForNewUnlock](#)
- [WaitForScreen](#)
- [WaitForUnlock](#)

Searches the text plane for the target string. If found, returns an OhioPosition object containing the target location. If not found, returns a null. The TargetString must be completely contained by the target area for the search to be successful. Null characters in the text plane are treated as blank spaces during search processing.

Visual Basic Syntax

```
[TTnbOhioPosition =] Screen.FindString(TargetString As String, startPos As OhioPosition, length As Integer, dir As OHIO_DIRECTION, IgnoreCase As Boolean)
```

Delphi Syntax

```
[TTnbOhioPosition :=] Screen.FindString(TargetString: string; startPos: TTnbOhioPosition; length: integer; dir: integer; IgnoreCase: boolean);
```

Parameters

Parameter	Description
<i>TargetString</i>	The target string.
<i>startPos</i>	The row and column where to start. The position is inclusive (for example, row 1, col 1 means that position 1,1 will be used as the starting location and 1,1 will be included in the search).
<i>length</i>	The length from startPos to include in the search.
<i>dir</i>	An OHIO_DIRECTION value.
<i>IgnoreCase</i>	Indicates whether the search is case sensitive. True means that case will be ignored. False means the search will be case sensitive.

Returns a character array containing the data from the Text, Color, Field or Extended plane of the virtual screen.

Visual Basic Syntax

[Char Array =] `Screen.GetData(start As TTnbOhioPosition, end As TTnbOhioPosition, plane As Integer)`

Delphi Syntax

[TCharArr :=] `Screen.GetData(startpos, endpos:TTnbOhioPosition; plane:integer);`

Parameters

Parameter	Description
<i>start</i>	The row and column where to start. The position is inclusive (for example, row 1, col 1 means that position 1,1 will be used as the starting location and 1,1 will be included in the data). "start" must be positionally less than "end".
<i>end</i>	The row and column where to end. The position is inclusive (for example, row 1, col 1 means that position 1,1 will be used as the ending location and 1,1 will be included in the data). "end" must be positionally greater than "start".
<i>plane</i>	A valid OHIO_PLANE value.

Updates the collection of OhioScreen objects. All OhioScreen objects that are available on the system at the time of the refresh will be added to the collection. Indexing of OhioScreen objects will not be preserved across refreshes.

Visual Basic Syntax

`Screen.Refresh`

Delphi Syntax

`Screen.Refresh`

Parameters

None.

The SendAid method sends an "aid" keystroke to the virtual screen. These aid keys can be thought of as special keystrokes, like the Enter key, the Tab key, or the Page Up key. All the valid special key values are contained in the OHIO_AID enumeration.

Visual Basic Syntax

`Screen.SendAid(aidkey As Integer)`

Delphi Syntax

```
Screen.SendAid(aidKey: integer);
```

Parameters

Parameters	Description
<i>aidKey</i>	The aid key to send to the virtual screen.

The SendKeys method sends a string of keys to the virtual screen. This method acts as if keystrokes were being typed from the keyboard.

The keystrokes will be sent to the location given. If no location is provided, the keystrokes will be sent to the current cursor location.

Visual Basic Syntax

```
Screen.SendKeys(text As String, location As OhioPosition)
```

Delphi Syntax

```
Screen.SendKeys(text: string; location: TTNbOhioPosition);
```

Parameters

Parameters	Description
<i>text</i>	The string of characters to be sent.
<i>location</i>	OhioPosition where the string should be written.

The SetString method sends a string to the virtual screen at the specified location. The string will overlay only unprotected fields, and any parts of the string which fall over protected fields will be discarded.

Visual Basic Syntax

```
Screen.SetString(text As String, location As TTNbOhioPosition)
```

Delphi Syntax

```
Screen.SetString(text: string; location: TTNbOhioPosition)
```

Parameters

Parameters	Description
<i>text</i>	String to place in the virtual screen
<i>location</i>	Position where the string should be written

General wait mechanism. This method basically waits until the mainframe application turns in an unlocked state and is able to receive input data.

Visual Basic Syntax

```
[Boolean] = Screen.Wait([Timeout] As Integer)
```

Delphi Syntax

```
[WordBool] = Screen.Wait([Timeout]: Integer)
```

Remarks

The Timeout parameter is optional and it is measured in milliseconds. This method returns True if the wait was successful and False if the timeout period has expired.

See Also

[WaitForConnect](#), [WaitForUnlock](#) and [WaitForNewScreen](#) methods.

This method waits for a screen containing the specified string. In the first case shown below, it returns True if the wait was successful and False if the Timeout period has expired. In the second case it receives an array of strings as parameter, and returns an integer indicating the number of the string (into the array of strings) that was found.

Visual Basic Syntax

```
[Boolean =] Screen.WaitFor(Value As String, [Timeout] As Integer, ResetWait As Boolean)
```

```
[Integer =] Screen.WaitFor(Values As String, [Timeout] As Integer, ResetWait As Boolean)
```

Delphi Syntax

```
[WordBool :=] Screen.WaitFor(Value: string; [Timeout]: Integer; ResetWait: Boolean);
```

```
[Integer :=] Screen.WaitFor(const Values: Array of string; [Timeout]: Integer; ResetWait: Boolean);
```

Parameters

Parameters	Description
<i>Value</i>	The string of characters expected to be found.
Values	The array of strings where one of them is expected to be found. In Visual Basic this is a

	string with several values separated by ";"
<i>TimeOut</i>	Optional parameter measured in milliseconds.
<i>ResetWait</i>	While this parameter is false, continue lopping until the expected string is found.

This method waits until a new screen arrives within the specified period of time. It is similar to `WaitForScreen` method except that it always waits for a new screen arrival.

This method returns `True` if the wait was successful and `False` if the `Timeout` period has expired.

Visual Basic Syntax

```
[Boolean =] Screen.WaitForNewScreen([TimeOut] As Integer)
```

Delphi Syntax

```
[WordBool :=] Screen.WaitForNewScreen([TimeOut]: Integer);
```

Parameters

Parameters	Description
<i>TimeOut</i>	Optional parameter measured in milliseconds.

This method waits until a new system unlock arrives within the specified period of time. It is similar to `WaitForUnlock` method except that it always waits for a new system unlock arrival. It returns `True` if the wait was successful and `False` if the `Timeout` period has expired.

Visual Basic Syntax

```
[Boolean =] Screen.WaitForNewUnlock([TimeOut] As Integer)
```

Delphi Syntax

```
[WordBool :=] Screen.WaitForNewUnlock([TimeOut]: Integer);
```

Parameters

Parameters	Description
<i>TimeOut</i>	Optional parameter measured in milliseconds.

This method waits for the first screen after a system lock. If the host system changes from an unlocked to a locked state (normally when we send an `Aid` key) the method waits until the first screen arrives. If `WaitForScreen` method is called after that event, it returns immediately. The method returns `True` if the wait was successful and `False` if the `Timeout` period has expired.

Visual Basic Syntax

[Boolean =] Screen.WaitForScreen([TimeOut] As Integer)

Delphi Syntax

[WordBool :=] Screen.WaitForScreen([TimeOut]: Integer);

Parameters

Parameters	Description
<i>TimeOut</i>	Optional parameter measured in milliseconds.

This method waits until a new system unlock arrives within the specified period of time. It is similar to WaitForUnlock method except that it always waits for a new system unlock arrival. This method returns True if the wait was successful and False if the Timeout period has expired.

Visual Basic Syntax

[Boolean =] Screen.WaitForUnlock([TimeOut] As Integer)

Delphi Syntax

[WordBool :=] Screen.WaitForUnlock([TimeOut]: Integer);

Parameters

Parameters	Description
<i>TimeOut</i>	Optional parameter measured in milliseconds.

5.3.4.1.6.3 Properties

- [Columns](#)
- [Cursor](#)
- [Fields](#)
- [OIA](#)
- [Rows](#)
- [String/Text](#)

Indicates the number of columns in the presentation space.

Visual Basic Syntax

[Integer =] Screen.Columns

Delphi Syntax

```
[Integer :=] Screen.Columns;
```

Remarks

This property returns an integer representing the number of columns on the current host screen.

Specifies the location of the cursor in the presentation space. The row and column of the cursor is contained within the *OhioPosition* object.

Visual Basic Syntax

```
Screen.Cursor [= TTnbOhioPosition]
```

Delphi Syntax

```
Screen.Cursor; [:= TTnbOhioPosition]
```

Remarks

This function returns an integer which represents the current cursor position, or sets the cursor position according to the *OhioPosition* passed as parameter.

Returns the *OhioFields* object associated with this presentation space. This provides another way to access the data in the virtual screen. The *OhioFields* object contains a snapshot of all the fields in the current virtual screen. Fields provide methods for interpreting the data in the non-text planes. Zero length fields (due to adjacent field attributes) are not returned in the *OhioFields* collection. For unformatted screens, the returned collection contains only one *OhioField* that contains the whole virtual screen.

Visual Basic Syntax

```
[TTnbOhioFields =] Screen.Fields
```

Delphi Syntax

```
[TTnbOhioFields :=] Screen.Fields;
```

Remarks

Returns the *Fields* object for the current host screen.

Returns the *OIA* object associated with this presentation space.

Visual Basic Syntax

```
[TTnbOhioOIA =] Screen.OIA
```

Delphi Syntax

```
[TTnbOhioOIA :=] Screen.OIA;
```

Remarks

This object can be used to query the status of the operator information area.

Indicates the number of rows in the presentation space.

Visual Basic Syntax

```
[Integer =] Screen.Rows
```

Delphi Syntax

```
[Integer :=] Screen.Rows;
```

Remarks

This property returns an integer representing the number of rows on the current host screen.

Returns the entire text plane of the virtual screen as a string.

Visual Basic Syntax

```
[String =] Screen.String
```

Delphi Syntax

```
[String :=] Screen.Text;
```

Remarks

All null characters and Field Attribute characters are returned as blank space characters.

In Delphi *String* is used as *Text* property because the name *String* can't be used as a function name.

5.3.4.1.7 OhioSession

The OhioSession represents a host session.

See Also

OhioSession [Properties](#), [Methods](#) and [Events](#).

5.3.4.1.7.1 Events

- [OnConnect](#)
- [OnConnectionFail](#)
- [OnDisconnect](#)
- [OnScreenChange](#)
- [OnSendAid](#)
- [OnSessionState](#)

Occurs after a successfully connection to the server is established.

Occurs after the connection to the server fails.

Occurs after a disconnection of the server.

This event is generated whenever the virtual screen is modified.

Occurs before an Aid key is to be sent.

Remarks

This event is fired when a [SendAid](#) or [SendKeys](#) methods are used to send an Aid Key and data to the mainframe. Occurs before the data is sent, allowing to take actions like filling edit fields or saving data typed in the emulator control.

Occurs when a session-state change takes place.

Remarks

A session state can be ssNoSession, ssSSCPLU and ssLULU. Any transition between the states fires this event.

5.3.4.1.7.2 Methods

- [Connect](#)
- [Disconnect](#)
- [WaitForConnect](#)
- [WaitForDisconnect](#)

Starts the communications link to the host.

Visual Basic Syntax

Session.Connect

Delphi Syntax

Session.Connect;

Parameters

None.

Stops the communications link to the host.

Visual Basic Syntax

Session.Disconnect

Delphi Syntax

Session.Disconnect;

Parameters

None.

This method waits until the telnet connection is successfully opened, or the timeout period expires.

Visual Basic Syntax

[Boolean =] Session.WaitForConnect([Timeout] As Integer)

Delphi Syntax

[WordBool : =] Session.WaitForConnect([Timeout]: Integer);

Remarks

The Timeout parameter is optional. This method returns True if the wait was successful and False if the Timeout period has expired. The Timeout parameter is measured in milliseconds.

If you pass a timeout value in the Timeout parameter of this method, the value of the property WaitTimeout will not have any effect.

This method waits until the telnet connection is successfully closed, or the timeout period expires.

Visual Basic Syntax

```
[Boolean =] Session.WaitForDisconnect([TimeOut] As Integer)
```

Delphi Syntax

```
[WordBool :=] Session.WaitForDisconnect([TimeOut]: Integer);
```

Remarks

The Timeout parameter is optional. This method returns True if the wait was successful and False if the Timeout period has expired. The TimeOut parameter is measured in milliseconds.

If you pass a timeout value in the Timeout parameter of this method, the value of the property WaitTimeout will not have any effect.

5.3.4.1.7.3 Properties

- [ConfigurationResource](#)
- [Connected](#)
- [Manager](#)
- [Screen](#)
- [SessionName](#)
- [SessionState](#)
- [SessionType](#)
- [Trace](#)

Indicates the ConfigurationResource for this OhioSession object.

Visual Basic Syntax

```
[String =] Session.ConfigurationResource
```

Delphi Syntax

```
[String :=] Session.ConfigurationResource;
```

Remarks

This property returns the full path of the configuration resource file for the current session.

Indicates whether this OhioSession object is connected to a host.

Visual Basic Syntax

[Boolean =] Session.Connected

Delphi Syntax

[Boolean :=] Session.Connected;

Remarks

True means connected, False means not connected.

Returns the OhioManager object for the current session.

Visual Basic Syntax

Session.Manager [= OhioManager]

Delphi Syntax

Session.Manager; [:= OhioManager]

Returns the OhioScreen object for the current session.

Visual Basic Syntax

[OhioScreen =] Session.Screen

Delphi Syntax

[OhioScreen :=] Session.Screen;

Returns the SessionName for this OhioSession object.

Visual Basic Syntax

[String =] Session.SessionName

Delphi Syntax

[String :=] Session.SessionName;

Remarks

The SessionName is unique among all instances of OhioSession.

Returns the SessionState for the Session object.

Visual Basic Syntax

[String =] Sessions.SessionState

Delphi Syntax

[String :=] Session.SessionState;

Returns the SessionType for this OhioSession object.

Visual Basic Syntax

[Integer =] Session.SessionType

Delphi Syntax

[Integer :=] Session.SessionType;

Sets the TNBXTTrace Control as its trace agent.

Visual Basic Syntax

Session.Trace [= *TTnbTrace*]

Delphi Syntax

Session.Trace; [*:= TTnbTrace*]

Remarks

You must set this property to get trace information from TNBXSsync control.

5.3.4.1.8 Syntax Conventions

The following text formats are used throughout the Components Reference:

For each property and method:

Italic text Denotes an object, in this case a TNBridge control.

Bold text Denotes a property or a method names.

[= value] or [:= value] Denotes values to be assigned.

[value =] or [value :=] Denotes return values.
 (TimeOut) Denotes a parameter to be passed to a method.
 ([TimeOut]) Denotes an optional parameter to be passed to a method.

Note: When we refer to *OhioObject* we are indicating that this object can be any of the objects that belong to the Ohio Base Class.

5.3.4.1.9 OhioSessions

Contains a collection of OhioSession objects. This list is a static snapshot of the list of OhioSession objects available at the time of the snapshot.

See Also

OhioSessions [Properties](#) and [Methods](#).

5.3.4.1.9.1 Methods

- [AddSession](#)
- [CloseSession](#)

Adds a session to the Sessions collection based on the specified configuration file.

Visual Basic Syntax

```
[TTnbOhioSession =] Sessions.AddSession(ConfigurationResource As String,
SessionName As String)
```

```
[TTnbOhioSession =] Sessions.AddSession(ConfigurationResource As Integer,
SessionName As String)
```

```
[TTnbOhioSession =] Sessions.AddSession(ConfigurationResource As TTnbComApi,
SessionName As String)
```

Delphi Syntax

```
[TTnbOhioSession :=] Sessions.AddSession(ConfigurationResource, SessionName:
string);
```

```
[TTnbOhioSession :=] Sessions.AddSession(ConfigurationResource: integer;
SessionName: string);
```

```
[TTnbOhioSession :=] Sessions.AddSession(ConfigurationResource: TTnbComApi;
SessionName: string);
```

Parameters

Parameters	Description
<i>ConfigurationResource</i>	Full path of the Configuration File (.ZCC) to use for

	host connection information. This parameter can be a string, an integer or a TtnbComApi type.
<i>SessionName</i>	Session's name.

Closes an OhioSession object. The OhioSession is considered invalid and is removed from the list of OhioSession objects. You can use this methods with the following syntaxes:

Visual Basic Syntax

```
Sessions.CloseSession(SessionObject As TtnbOhioSession)
```

```
Sessions.CloseSession(SessionName As String)
```

Delphi Syntax

```
Sessions.CloseSession(SessionObject: TtnbOhioSession);
```

```
Sessions.CloseSession(SessionName: string);
```

Parameters

Parameters	Description
<i>SessionObject</i>	The OhioSession to close.
<i>SessionName</i>	The SessionName of the OhioSession to close.

5.3.4.1.9.2 Properties

- [Count](#)
- [Item](#)
- [Manager](#)

Returns the number of OhioSessions objects contained in this collection (the number of Sessions currently opened).

Visual Basic Syntax

```
[Integer =] Sessions.Count
```

Delphi Syntax

```
[Integer :=] Sessions.Count;
```

Returns the OhioSessions object at the given index. "One based" indexing is used in all Ohio collections. For example, the first OhioSession in this collection is at index 1.

Visual Basic Syntax

[TTnbOhioSession] = Sessions.Item(index As Variant)

Delphi Syntax

[TTnbOhioSession] := Sessions.Item(index: Variant);

Parameters

Parameters	Description
<i>index</i>	The index of the target OhioSession.

Returns the OhioManager object for the current session.

Visual Basic Syntax

Sessions.Manager [= OhioManager]

Delphi Syntax

Sessions.Manager; [: = OhioManager]

5.3.4.2 Constants

- [Ohio Constants](#)
- [3270 Key Values](#)
- [5250 Key Values](#)

5.3.4.2.1 3270 Key Values

Aid Key Constants	Value
OHIO_AID_KEY_ATTN	\$0182
OHIO_AID_KEY_BACKSPACE	\$0183
OHIO_AID_KEY_BACKTAB	\$0184
OHIO_AID_KEY_CLEAR	\$0186
OHIO_AID_KEY_CURSORSEL	\$0188
OHIO_AID_KEY_DELETE	\$0189
OHIO_AID_KEY_DOWN	\$018B
OHIO_AID_KEY_DUP	\$018C
OHIO_AID_KEY_ENTER	\$018D
OHIO_AID_KEY_ERASEEOF	\$018E
OHIO_AID_KEY_ERASEINPUT	\$018F
OHIO_AID_KEY_HOME	\$0190
OHIO_AID_KEY_LEFT	\$0193
OHIO_AID_KEY_MARK	\$0194
OHIO_AID_KEY_NEWLINE	\$0195
OHIO_AID_KEY_PA1	\$0196
OHIO_AID_KEY_PA2	\$0197
OHIO_AID_KEY_PA3	\$0198

OHIO_AID_KEY_PF1	\$01A0
OHIO_AID_KEY_PF2	\$01A1
OHIO_AID_KEY_PF3	\$01A2
OHIO_AID_KEY_PF4	\$01A3
OHIO_AID_KEY_PF5	\$01A4
OHIO_AID_KEY_PF6	\$01A5
OHIO_AID_KEY_PF7	\$01A6
OHIO_AID_KEY_PF8	\$01A7
OHIO_AID_KEY_PF9	\$01A8
OHIO_AID_KEY_PF10	\$01A9
OHIO_AID_KEY_PF11	\$01AA
OHIO_AID_KEY_PF12	\$01AB
OHIO_AID_KEY_PF13	\$01AC
OHIO_AID_KEY_PF14	\$01AD
OHIO_AID_KEY_PF15	\$01AE
OHIO_AID_KEY_PF16	\$01AF
OHIO_AID_KEY_PF17	\$01B0
OHIO_AID_KEY_PF18	\$01B1
OHIO_AID_KEY_PF19	\$01B2
OHIO_AID_KEY_PF20	\$01B3
OHIO_AID_KEY_PF21	\$01B4
OHIO_AID_KEY_PF22	\$01B5
OHIO_AID_KEY_PF23	\$01B6
OHIO_AID_KEY_PF24	\$01B7
OHIO_AID_KEY_RESET	\$019A
OHIO_AID_KEY_RIGHT	\$019B
OHIO_AID_KEY_SYSREQ	\$019C
OHIO_AID_KEY_TABFORWARD	\$019D
OHIO_AID_KEY_UP	\$019F
OHIO_AID_KEY_PRINT	\$01C0
OHIO_AID_KEY_PGUP	\$01C1
OHIO_AID_KEY_PGDOWN	\$01C2

5.3.4.2.2 5250 Key Values

Aid Key Constants	Value
OHIO_AID_KEY_ATTN	\$0182
OHIO_AID_KEY_BACKSPACE	\$0183
OHIO_AID_KEY_BACKTAB	\$0184
OHIO_AID_KEY_CLEAR	\$0186
OHIO_AID_KEY_CURSORSEL	\$0188
OHIO_AID_KEY_DELETE	\$0189
OHIO_AID_KEY_DOWN	\$018B
OHIO_AID_KEY_DUP	\$018C
OHIO_AID_KEY_ENTER	\$018D
OHIO_AID_KEY_ERASEEOF	\$018E
OHIO_AID_KEY_ERASEINPUT	\$018F
OHIO_AID_KEY_HOME	\$0190
OHIO_AID_KEY_LEFT	\$0193
OHIO_AID_KEY_MARK	\$0194

OHIO_AID_KEY_NEWLINE	\$0195
OHIO_AID_KEY_PA1	\$0196
OHIO_AID_KEY_PA2	\$0197
OHIO_AID_KEY_PA3	\$0198
OHIO_AID_KEY_PF1	\$01A0
OHIO_AID_KEY_PF2	\$01A1
OHIO_AID_KEY_PF3	\$01A2
OHIO_AID_KEY_PF4	\$01A3
OHIO_AID_KEY_PF5	\$01A4
OHIO_AID_KEY_PF6	\$01A5
OHIO_AID_KEY_PF7	\$01A6
OHIO_AID_KEY_PF8	\$01A7
OHIO_AID_KEY_PF9	\$01A8
OHIO_AID_KEY_PF10	\$01A9
OHIO_AID_KEY_PF11	\$01AA

5.3.4.2.3 Ohio Constants

Constants for all Ohio classes.

OHIO_INFO

Constant	Value	Description
OHIO_VERSION_LEVEL	OHIO 1.00	The OHIO version level of this implementation. The form is "OHIO nn.nn"
OHIO_VENDOR_NAME	Cybele Software Inc.	The name of the vendor providing this OHIO implementation. Format is vendor defined.
OHIO_VENDOR_PRODUCT_VERSION	TN Bridge Integration Pack 3.0	The vendor product version that is providing the OHIO implementation. Format is vendor specific.

OHIO_SESSION cfg keys

Constant	Value	Description
OHIO_SESSION_TYPE	OHIO_SESSION_TYPE=	The OHIO Session Type: unknown host, 3270 or 5250 host.
OHIO_SESSION_HOST	OHIO_SESSION_HOST=	The OHIO Session Host.
OHIO_SESSION_PORT	OHIO_SESSION_PORT=	The OHIO Session Port.
OHIO_SESSION_NAME	OHIO_SESSION_NAME=	The OHIO Session Name.
OHIO_SESSION_CONFIG_RESOURCE	OHIO_SESSION_CONFIG_RESOURCE=	The OHIO Session Configuration Resource.
OHIO_SESSION_TN_ENHANCED	OHIO_SESSION_TN_ENHANCED=	
OHIO_SESSION_SCREEN_SIZE	OHIO_SESSION_SCREEN_SIZE=	The Screen Size for the OHIO Session.
OHIO_SESSION_CODE_PAGE	OHIO_SESSION_CODE_PAGE=	The Code Page for the OHIO Session.

OHIO_DIRECTION

Constant	Value	Description
OHIO_DIRECTION_FORWARD	0	Forward (beginning towards end)
OHIO_DIRECTION_BACKWARD	1	Backward (end towards beginning)

OHIO_TYPE

Constant	Value	Description
OHIO_TYPE_UNKNOWN	0	Unknown Host
OHIO_TYPE_3270	1	3270 Host
OHIO_TYPE_5250	2	5250 Host

OHIO_STATE

Constant	Value	Description
OHIO_STATE_DISCONNECTED	0	The communication link to the host is disconnected
OHIO_STATE_CONNECTED	1	The communication link to the host is connected

OHIO_PLANE

Constant	Value	Description
OHIO_PLANE_TEXT	1	Indicates Plane Text (character data)
OHIO_PLANE_COLOR	2	Indicates Plane Color (standard HLLAPI CGA color values)
OHIO_PLANE_FIELD	4	Indicates Field Attribute Plane (field attribute bytes)
OHIO_PLANE_EXTENDED	8	Indicates Extended Plane (extended attribute bytes)

OHIO_COLOR

Constant	Value	Description
OHIO_COLOR_BLACK	0	Black
OHIO_COLOR_BLUE	1	Blue
OHIO_COLOR_GREEN	2	Green
OHIO_COLOR_CYAN	3	Cyan
OHIO_COLOR_RED	4	Red
OHIO_COLOR_MAGENTA	5	Magenta
OHIO_COLOR_YELLOW	6	Yellow
OHIO_COLOR_WHITE	7	White

OHIO_EXTENDED

Constant	Value	Description
OHIO_EXTENDED_HILITE	\$C0	Bitmask for Highlighting Bits
OHIO_EXTENDED_COLOR	\$38	Bitmask for Color Bits
OHIO_EXTENDED_RESERVED	\$07	Bitmask for Reserved Bits
OHIO_EXTENDED_HILITE_NORMAL	\$00	Normal highlighting
OHIO_EXTENDED_HILITE_BLINK	\$40	Blinking highlighting
OHIO_EXTENDED_HILITE_REVERSEVIDEO	\$80	Reverse video highlighting
OHIO_EXTENDED_HILITE_UNDERSCORE	\$C0	Under score highlighting
OHIO_EXTENDED_COLOR_DEFAULT	\$00	Default color
OHIO_EXTENDED_COLOR_BLUE	\$08	Blue
OHIO_EXTENDED_COLOR_RED	\$10	Red
OHIO_EXTENDED_COLOR_PINK	\$18	Pink
OHIO_EXTENDED_COLOR_GREEN	\$20	Green
OHIO_EXTENDED_COLOR_TURQUOISE	\$28	Turquoise
OHIO_EXTENDED_COLOR_YELLOW	\$30	Yellow
OHIO_EXTENDED_COLOR_WHITE	\$38	White
OHIO_EXTENDED_COLUMN_SEPARATOR	\$02	Separator

OHIO_FIELD

Constant	Value	Description
OHIO_FIELD_ATTRIBUTE	\$C0	Bitmask for field attribute
OHIO_FIELD_PROTECTED	\$20	Protected field
OHIO_FIELD_NUMERIC	\$10	Numeric field
OHIO_FIELD_HIDDEN	\$0C	Hidden field
OHIO_FIELD_PEN_SELECTABLE	\$08	Pen selectable field
OHIO_FIELD_HIGH_INTENSITY	\$04	High Intensity field
OHIO_FIELD_RESERVED	\$02	Reserved field
OHIO_FIELD_MODIFIED	\$01	Modified field

OHIO_UPDATE

Constant	Value	Description
OHIO_UPDATE_CLIENT	0	Update initiated by client
OHIO_UPDATE_HOST	1	Update initiated by host

OHIO_OWNER

Constant	Value	Description
OHIO_OWNER_UNKNOWN	0	Uninitialized

OHIO_OWNER_APP	1	Application or 5250 host
OHIO_OWNER_MYJOB	2	3270 - Myjob
OHIO_OWNER_NVT	3	NVT (Network Virtual Terminal) mode
OHIO_OWNER_UNOWNED	4	Unowned (3270 only)
OHIO_OWNER_SSCP	5	SSCP (3270 only)

OHIO_INPUTINHIBITED

Constant	Value	Description
OHIO_INPUTINHIBITED_NOTINHIBITED	0	Input not inhibited
OHIO_INPUTINHIBITED_SYSTEM_WAIT	1	Input inhibited by a System Wait state ("X SYSTEM" or "X []")
OHIO_INPUTINHIBITED_COMMCHECK	2	
OHIO_INPUTINHIBITED_PROGCHECK	3	
OHIO_INPUTINHIBITED_MACHINE_CHECK	4	
OHIO_INPUTINHIBITED_OTHER	5	Input inhibited by something other than above states

OHIO_SESSION_STATE

Constant	Value	Description
OHIO_SESSION_STATE_NOSESSION	0	No session established
OHIO_SESSION_STATE_SSCPLU	1	SSCPLU session state
OHIO_SESSION_STATE_LULU	2	LULU session state

OHIO_EXCEPTION

Constant	Value	Description
OHIO_EXCEPTION_SESSION_ALREADY_EXISTS		Already exists a session with that name
OHIO_EXCEPTION_BADCFGFILE		Error in configuration file
OHIO_EXCEPTION_BADHOSTNAME		There is no data provided in '+OHIO_SESSION_HOST+' configuration field
OHIO_EXCEPTION_BADPORTNUMBER		Data provided in '+OHIO_SESSION_PORT+' configuration field is not a number
OHIO_EXCEPTION_INVALIDPORTNUMBER		Data provided in '+OHIO_SESSION_PORT+' configuration field is out of range
OHIO_EXCEPTION_TOOMANYSESSIONS		Too many sessions

5.3.4.3 OHIO Base Class

This is the base class for all Ohio classes. It contains all common Ohio methods and properties.

See Also

Ohio Base Class [Properties](#) and [Methods](#).

5.3.4.3.1 Methods

- [CreateOhioPosition](#)

5.3.4.3.1.1 CreateOhioPosition

Creates an OhioPosition object.

Visual Basic Syntax

```
[TTnbOhioPosition =] OhioObject.CreateOhioPosition(row As Integer, col As Integer)
```

Delphi Syntax

```
[TTnbOhioPosition :=] OhioObject.CreateOhioPosition(row,col: integer);
```

Parameters

Parameter	Description
<i>row</i>	The row coordinate
<i>col</i>	The column coordinate

5.3.4.3.2 Properties

- [OhioVersion](#)
- [VendorName](#)
- [VendorObject](#)
- [VendorProductVersion](#)

5.3.4.3.2.1 OhioVersion

Returns the Ohio Version level for this implementation (OHIO 1.00).

Visual Basic Syntax

```
[String =] OhioObject.OhioVersion
```

Delphi Syntax

```
[String :=] OhioObject.OhioVersion;
```

5.3.4.3.2.2 VendorName

Returns the name of the vendor providing this OHIO implementation. Format is vendor defined.

Visual Basic Syntax

```
[String =] OhioObject.VendorName
```

Delphi Syntax

```
[String :=] OhioObject.VendorName;
```

5.3.4.3.2.3 VendorObject

Returns the VendorObject for the current session. This property returns different objects according to the object who calls it.

Visual Basic Syntax

```
[TTnbTelnet =] Screen.VendorObject
```

```
[TTnbTelnet =] Session.VendorObject
```

```
[TTnbOIA =] OIA.VendorObject
```

```
[TTnbSessions =] Sessions.VendorObject
```

```
[TTnbField =] Field.VendorObject
```

Delphi Syntax

```
[TTnbTelnet :=] Screen.VendorObject;
```

```
[TTnbTelnet :=] Session.VendorObject;
```

```
[TTnbOIA :=] OIA.VendorObject;
```

```
[TTnbSessions :=] Sessions.VendorObject;
```

```
[TTnbField :=] Field.VendorObject;
```

5.3.4.3.2.4 VendorProductVersion

Indicates the vendor product version that is providing the OHIO implementation. Format is vendor specific.

Visual Basic Syntax

[String =] *OhioObject*.VendorProductVersion

Delphi Syntax

[String :=] *OhioObject*.VendorProductVersion;

5.3.4.3.3 Syntax Conventions

The following text formats are used throughout the Components Reference:

For each property and method:

Italic text Denotes an object, in this case a TNBridge control.

Bold text Denotes a property or a method names.

[= value] or [:= value] Denotes values to be assigned.

[value =] or [value :=] Denotes return values.

(TimeOut) Denotes a parameter to be passed to a method.

([TimeOut]) Denotes an optional parameter to be passed to a method.

Note: When we refer to *OhioObject* we are indicating that this object can be any of the objects that belong to the Ohio Base Class.

5.4 DDE Reference

The following Dynamic Data Exchange (DDE) functions are supported:

Connect

Topic=*z/Scope's exe name* [zClassic, zExpress3270, etc], Service=*short session name* (A, B, etc)

Request

Item=PS

returns the presentation space (screen text).

Item=Cursor

returns the cursor position (relative to 1).

Item=Rows

returns the number of rows in the presentation space.

Item=Columns

returns the number of columns in the presentation space.

Item=Emulator

returns the window handle of the z/Scope window.

Item=Keyboard

returns "Locked" if the keyboard is locked or else it returns "Clear".

Item=OIA

returns the Operator Information Area (note that this is NOT in IBM format).

Poke

Item=Keystroke

sends one or more keystrokes to z/Scope (same format as WinHLLAPI). See the [Keystroke Table](#) for a list of all the keystrokes.

Item=Cursor

moves the cursor position (relative to 1).

Item=EscChar

sets the escape character for Item=Keystroke (default is '@').

5.4.1 DDE Keystroke Table

Keystroke Table:

Host Function Key	Mnemonic*	5250	3270
@	@@	x	x
Alt	@A		
Attention	@A@Q	x	x
Backspace	@<	x	x
Backtab (Left Tab)	@B	x	x
Clear	@C	x	x
Cursor Down	@V	x	x
Cursor Left	@L	x	x
Cursor Right	@Z	x	x
Cursor Select	@A@J	x	x
Cursor Up	@U	x	x
Delete	@D	x	x
Dup	@S@x	x	x
End	@q	x	x
Enter	@E	x	x
Erase EOF	@F	x	x
Erase Input	@A@F	x	x
Field Exit	@A@E	x	
Field Mark	@S@y	x	
Field -	@A@-	x	
Field +	@A@+	x	
Help	@H	x	
Home	@0 (zero)	x	x
Insert	@I	x	x
Insert Toggle	@A@I	x	x
Host Print	@P	x	
Left Tab (Backtab)	@B	x	x
New Line	@N	x	x
Page Up	@u	x	x
Page Down	@v	x	x
Record Backspace	@A@<	x	
Right Tab (Tab)	@T	x	x
Shift	@S		
Sys Request	@A@H	x	x
Tab (Right Tab)	@T	x	x
Test	@A@C	x	x
PA1	@x	x	
PA2	@y	x	
PA3	@z	x	

PF1/F1	@1	X	X
PF2/F2	@2	X	X
PF3/F3	@3	X	X
PF4/F4	@4	X	X
PF5/F5	@5	X	X
PF6/F6	@6	X	X
PF7/F7	@7	X	X
PF8/F8	@8	X	X
PF9/F9	@9	X	X
PF10/F10	@a	X	X
PF11/F11	@b	X	X
PF12/F12	@c	X	X
PF13	@d	X	X
PF14	@e	X	X
PF15	@f	X	X
PF16	@g	X	X
PF17	@h	X	X
PF18	@i	X	X
PF19	@j	X	X
PF20	@k	X	X
PF21	@l	X	X
PF22	@m	X	X
PF23	@n	X	X
PF24	@o	X	X

* Note: The DDE keystroke mnemonics are case sensitive!

6 Appendix A - Character Conversion Tables

Z/Scope is shipped with a large list of EBCDIC to ASCII internal conversion tables. It also allows you to use user-generated external conversion tables.

For more information on how to create and apply your own character conversion tables, see the following sections:

- [Internal Conversion Tables](#)
- [Modifying the Character Table](#)
- [Using an external Character Table](#)

Related Topics

- [Selecting a different Character Table](#)

6.1 Internal Conversion Tables

The following table lists the internal ASCII/EBCDIC conversion tables included in z/Scope:

Country	Code Page
Albania	CP500
Arabic Countries	CP420
Australia	CP037
Austria Euro	CP273
Austria	CP1141
Belarus Euro	CP1154
Belarus	CP1025
Belarus/EEUU	CP037
Belgium Euro	CP1148
Belgium	CP500
Belgium	CP037
Bosnia/Herzegovina	CP870
Brazil Euro	CP1140
Brazil	CP037
Bulgaria Euro	CP1154
Bulgaria	CP1125
Canada Euro	CP1140
Canada	CP037
Canada	CP1047
Croatia Euro	CP1153
Croatia	CP870
Czech Euro	CP1153
Czech	CP870
Denmark Euro	CP1142
Denmark	CP227
Estonia Euro	CP1157
Estonia	CP1122
Estonia	CP037
Finland Euro	CP1143
Finland	CP278
France Euro	CP1143
France	CP297
Germany Euro	CP1141
Germany	CP273
Greece Euro	CP874
Greece	CP423
Greece	CP875
Hebrew	CP424
Hungary euro	CP1153
Hungary	CP870

International	CP850
Iceland Euro	CP1149
Iceland	CP871
Italy Euro	CP1144
Italy	CP280
Latin America Euro	CP1145
Latin America	CP284
Latvia Euro	CP1156
Latvia	CP1112
Latvia	CP037
Lithuania Euro	CP1156
Lithuania	CP1112
Lithuania	CP037
Macedonia Euro	CP1154
Macedonia	CP1025
Multinational Euro	CP1148
Multinational ISO Euro	CP924
Multinational	CP500
Netherlands Euro	CP1140
Netherlands	CP037
Norway Euro	CP1142
Norway	CP277
Poland Euro	CP1153
Poland	CP870
Portugal Euro	CP1140
Portugal	CP037
Romania Euro	CP1153
Romania	CP870
Russian Euro	CP1154
Russian	CP1025
Serbia Cyrillic Euro	CP1154
Serbia Latin Euro	CP1153
Serbia/Montenegro	CP1025
Slovenia Euro	CP1153
Slovenia	CP870
Slovakia Euro	CP1153
Slovakia	CP870
Spain Euro	CP1145
Spain	CP284
Sweden Euro	CP1143
Sweden	CP287
Switzerland Euro	CP1148
Switzerland	CP500
Turkey Euro	CP1155
Turkey	CP1026
Ukraine Euro	CP1158
Ukraine	CP1123

You can save the character conversion modifications you have made to an external character conversion file (.ebc) so that it can be used in the future. Just select the 'Save Ebcdic Table' option from the 'Session' menu and type a name for the conversion table file.



To learn how to load a previously saved Conversion Table, see the [next topic](#).

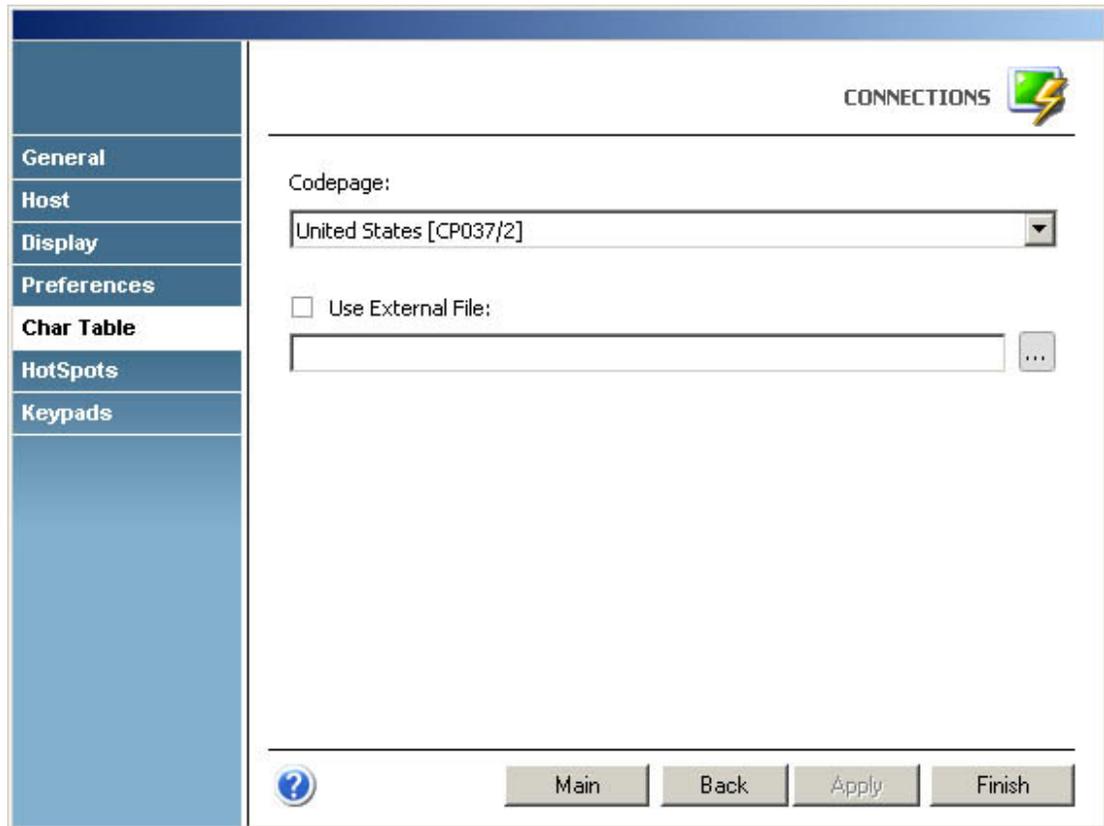
6.3 Using an external Character Table

After saving your own character table to a file, you can load it from within any connection. To do this, follow these steps:

1. Go to the 'Settings' dialog and click on the 'Connections' icon.



2. Select the 'Char Table' tab.
3. Select the 'Use external file' option and type the full path of the (.ebc) file to use.



4. Click on 'Finish'.

The next time you connect, the specified external conversion table will be loaded for the selected connection.

7 Appendix B - Regular Expressions

Regular Expressions can be considered a programming language that was specifically designed for string processing. Its main purpose is to locate patterns of substrings within a larger string, according to relative position, context, case and many other attributes.

To achieve this, the Regular Expressions language recognizes a set of special characters that can be compared in functionality to the wildcard characters * and ? in the DOS environment. The language employs many of these special characters to provide endless possibilities when searching for a certain pattern within a string. There's also a system for grouping parts of substrings and intermediate results during a search operation.

z/Scope takes advantage of the power and simplicity of the Regular Expressions language for defining Hotspots, one of z/Scopes' key features. The creation of a Hotspot requires the user to specify the criteria that a text string in the emulation display must fulfill in order to be recognized and respond to mouse clicks. See [Creating/Editing a Hotspot](#).

Most letters and characters will simply match themselves. For example, the regular expression "engine" will match the string "engine" exactly. However, there are some special characters (usually called metacharacters) that do not match themselves. Instead, they are used to define rules and patterns that will be looked for when analyzing the strings.

Here's a comprehensive list of all available metacharacters:

Character	Description	Example
^	Matches the position at the beginning of the string.	^B matches "B" but only if it is the first character in the string.
\$	Matches the position at the end of the string.	\$p matches "p" but only if it is the last character in the string.
.	Matches any single character.	le. matches "leg" and "let".
+	Matches the preceding character 1 or more times.	ca+t matches "cat" and "caat" but not "ct".
*	Matches the preceding character 0 or more times.	ca*t matches "ct", "cat", "caat" and so on.
?	Matches the preceding character 0 or 1 times.	si?t matches "st" and "sit" only.
[xyz]	Matches any one of the enclosed characters (character set).	[gdp]ot matches the "got", "dot" and "pot".
[^xyz]	Matches any character not enclosed (complementary set).	[^aeiou] matches any character that is not a vocal.
[x y]	Matches either x or y.	wom[a e]n matches either "woman" or "women".
[a-z]	Matches any character in the specified range (character range).	[a-z] matches any lowercase letter of the alphabet.
[^a-z]	Matches any character not in the specified range (complementary range).	[^a-z] matches any character that is not in the alphabet.
\b	Matches a word boundary (the position between a word and a space).	a\b matches the "a" in "general" but not the "a" in "fall".
\B	Matches a nonword boundary.	a\B matches the "a" in "fall" but not the "a" in "general".
\s	Matches any white space character including space, tab, form-feed, and so on.	
\S	Matches any non-white space character.	

\d	Matches a digit character. Equivalent to [0-9].	
\D	Matches any non-digit character. Equivalent to [^0-9].	
\w	Matches any word character including underscore. Equivalent to [A-Za-z0-9_].	
\W	Matches any non-word character. Equivalent to [^A-Za-z0-9_].	
{n}	Matches a character exactly n times.	p{2} does not match the "p" in "peach" but matches the two p's in "apple".
{n,}	Matches a character at least n times.	p{2,} does not match the "p" in "peach" and matches all the p's in "apppp".
{n,m}	Matches a character at least n and at most m times.	p{1,3} matches the first three p's in "appppp".

If you need to search for one of the characters that are reserved as metacharacters, you can do so by placing a backslash (\) before the desired character. In this way, for example, \? will actually match "?" instead of matching the position at the end of the string.

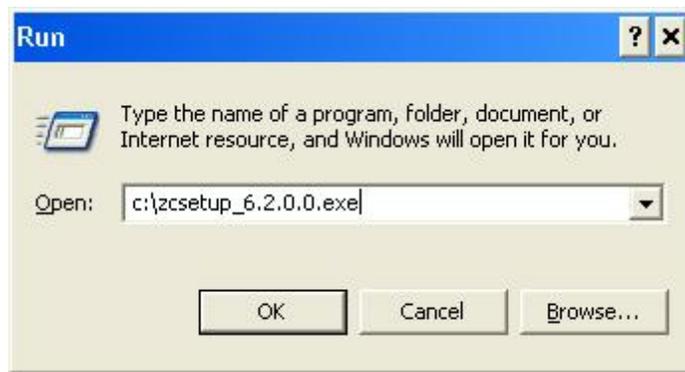
Related Topics

- [Creating/Editing a Hotspot](#)

8 Appendix C - Useful setup installation command-line switches

Z/Scope installation for 6.2 edition can be run in 'silent' mode, that is, without the need for user interaction. This can be useful if you are a system administrator and you want to automate the z/Scope installation or if you are deploying it over your local network.

To do this, you can click the 'Start' button in your Windows desktop and then select the 'Run' option. The following dialog will be displayed:



Use the 'Browse' button to locate z/Scope installation file and then add the following switches to the path outside the ""

/S

For example: "c:\zcsetup_6.2.0.0.exe" /S

This will install z/Scope in full silent mode.

To install z/Scope for all the Windows' users in the computer, use the command:

/ALLUSERS

For example: "c:\zcsetup_6.2.0.0.exe" /ALLUSERS

This will create a shortcut on every user's desktop and start menu.

NOTE: Each new command can be added to complete the full installation configuration:
For example: "c:\zcsetup_6.2.0.0.exe" /S /ALLUSERS /USER="John Doe" /SERIAL="XXXX-XXXX-XXXX-XXXX"

OTHER COMMANDS

- To pass the license information use the following parameters:

/USER="..."
/SERIAL="..."

Enter between the "..." your username and your serial key. For example:

/USER="John Doe" /SERIAL="XXXX-XXXX-XXXX-XXXX"

- You can block plugins during installation with the following command

/BLOCKPLUGINS="..."

Accepted plugin values are the following: 3270, 5250, VT, FTP, Printer, Favorites,

Browser, Start Panel, Settings. Separate the values with a comma (,) and don't use spaces. For example:

```
/BLOCKPLUGINS="Browser,5250"
```

To learn more about plugins, see the [Plugin Appendix](#).

A good complement to the BLOCKPLUGINS parameter is:

```
/HIDESTARTPAGE
```

Adding this parameter in the command line will make the start page not visible by default. Learn how to configure this after the installation in Start Page Configuration.

- Another useful functionality that z/Scope provides is the possibility to pass configuration files together with the installation through command line. Thus the administrator can configure z/Scope and establish a default configuration for all the users.

The configuration files are stored in

```
C:\Documents and Settings\(user)\Application Data\Cybele Software\zScope6
```

and they are the following:

- connections.xml: This file holds the list of configured connections and their settings.
- favorites.xml: This file holds the list of favorites with their names and folder tree.
- hotspots.xml: This file holds the list of HotSpots and their settings.
- screen.xml: This file holds the list of Screen Styles and their settings
- the environment file: This file holds all the general settings for z/Scope, that you will find in Options - Configuration Settings - Environment. Each z/Scope edition has a different environment file
 - zclassic.environment.xml for the Classic edition
 - zworkbench.environment.xml for the Workbench edition
 - zexpress3270.environment.xml for the Express 3270 edition
 - zexpress5250.environment.xml for the Express 5250 edition
 - zexpressvt.environment.xml for the Express VT edition

When passing configuration files as command line parameters, you should rename each (name).xml file to (name).default.xml, except for the environment file which keeps its original name.

The command parameter is:

```
/FILES="..."
```

In between the "..." goes the file name with its full path. If the files are located in the same directory as the installation package, you can replace the path by .\

The files can be located in the same computer where it will be installed or in a computer in the local network.

Separate the values with a comma (,) and don't use spaces. For example:

```
/FILES=".\\favorites.default.xml,c:\\Installer\\zexpress3270.environment.default.xml,\\company-server\\Shared\\connections.default.xml"
```

- z/Scope gives you the possibility to choose where to store your configuration files upon installation. To do so, use the following parameter:

```
/CONFIGPATH="..."
```

Using this parameter will set the path enclosed in between " " as the default path for configuration for all windows users accessing this z/Scope installation. To see how you can change this path inside z/Scope go to [Configuration Setup Settings](#).

- If you are using the Remote Configuration Server, you probably want to deploy the application for end-user already pointing to the correct URL, especially if you are using User Profiles. In this case, the parameter you need to add to the command line is:

```
/REMOTECONFIG="..."
```

Enclosed in between "..." specify the URL where the Remote Configuration Server is placed. For example:

```
/REMOTECONFIG="http://127.0.0.1:8089/zScopeRCS"
```

9 Appendix D - Web Server Configuration

For z/Scope Web-to-Host Edition only.

z/Scope Web-to-Host can be installed in a Web Server so users can run instances of it in their local workstations inside an ActiveX enabled or Mozilla web browser.

Three base configuration files are copied to the web folder upon installation:

1 zReg6.txt: This file contains licensing registration parameters. z/Scope Web-to-Host allows two modes of licensing registration: on client desktop and on web-server.

```
[Registration]
Username=
SerialNumber=
BlockTrials=FALSE
AllowRegistration=TRUE
GetAKeyURL=http://www.cybelesoft.com/
```

Registration section

Username : Licensed name.
SerialNumber : Licensed serial number.
BlockTrials : Avoids the use of the z/Scope in trial mode.

AllowRegistration : Enables/disables entering individual licensing registration in client web browser.

GetAKeyURL : Allows to redirect the "Get a Key" button to any url.

2 zUpd6.txt: This file contains version information. This allows you to manage automated web update to client desktops.

```
[Update]
Version=6,2,0,0
Location=zQ6.bin
CloseOnInstall=true
Mandatory=false
```

Update section

Version : Specifies the zQ6.bin file internal version. (This information is provided by Cybele Software Inc.)

Location : Specifies the location of zQ6.bin file.

CloseOnInstall : Specifies whether the update dialog will close itself automatically or present a "Continue" button upon finalization of install process.

Mandatory : Specifies whether the update is automatically started or if the user is prompted when a new version is available.

3 zCfg6.txt: This file contains z/Scope configuration parameters:

```
[Files]
Connections=connections.xml
Screenstyles=screen.xml
Keyboard=keyboard.xml
Macros=macros.xml
Hotspots=hotspots.xml
```

```
[Configuration]
Language=1
HelpUrl=Help/index.html
HideMenubar=false
HideToolbar=false
```

Files section

Connections : Points to the xml file where the shared connections are stored.

ScreenStyles : Points to the xml file where the shared Screen Styles are stored.

Keyboard : Points to the xml file where the shared keyboard map are stored.

Macros : Points to the xml file where the shared macros are stored.

Hotspots : Points to the xml file where the shared HotSpots are stored.

Configuration section

Language : Language number. 1. English 2. Spanish
HelpUrl : Points to the index of the html help.
HideMenubar : Allows to hide the z/Scope menu bar.
HideToolbar : Allows to hide the z/Scope toolbar.

10 Appendix E - Plugins

Z/Scope enables you to activate or deactivate several part of the system, to personalize the interface and adjust z/Scope to your specific needs.

To activate or deactivate plugins visit the following section:

- [Plugins Configuration](#)

Below is a list of all the current plugins available, together with an explanation:

3270

This plugin enables access to all connections of the tn3270 protocol type. By deactivating this plugin you will not be able to create or view connections of this type. It will also hide 3270 connections marked as Favorites in the Favorites list.

5250

This plugin enables access to all connections of the tn5250 protocol type. By deactivating this plugin you will not be able to create or view connections of this type. It will also hide 5250 connections marked as Favorites in the Favorites list.

VT

This plugin enables access to all connections of the vt protocol type. By deactivating this plugin you will not be able to create or view connections of this type. It will also hide vt connections marked as Favorites in the Favorites list.

FTP

This plugin enables access to FTP connections. By deactivating this plugin you will not be able to create or view connections of this type. It will also hide FTP connections marked as Favorites in the Favorites list.

Printer

This plugin enables the printer emulation for tn3270 or tn5250 hosts. By disabling this plugin you will not be able to view printer-only connections; you will not be able to emulate the printer for printer-and-display connections type; and you will not be able to configure printer-only or printer-and-display connections type.

Favorites

This plugin gives you access to the Favorites list. By disabling this plugin you will not

be able to visualize the Favorites list or add connections, web sites or Feeds to it. As a side effect, you would be deactivating the Browser plugin with it, since the browser can only be accessed from the Favorites list for the moment.

Browser

This plugin allows you to view web sites without leaving the z/Scope interface. By disabling this plugin you will not be able to access any web page from the z/Scope interface. It will also hide web pages marked as Favorites in the Favorites list.

Start Panel

This plugin allows you to be able to see the Start Page at your convenience. Deactivating this plugin will hide the start page not only on start but during all the time the plugin is disabled.

Settings

This plugin allows you to see configuration options. Deactivating this plugin will block access to all the configuration. This action cannot be reversed and it's intended for use only through SMS to restrict user access to options.

Editor

This plugin allows you to be able to see and use Code Editor options when using z/Scope Workbench. The code Editor for Mainframe, AS/400 and Unix files supports JCL, RPG/IV (also FREE & SQL), COBOL, CL, ASM370, CLIST, DDS, REXX, MACRO ASSEMBLER, JAVA, PHP, HTML, XML, ASP, C/C++ and other programming languages.

NOTE: Deactivation of a plugin does not mean that the information they hold disappears, only that it becomes invisible to the user until it's activated again.

11 Purchasing z/Scope

By purchasing any edition of z/Scope Terminal Emulator you will access to technical support, free upgrades and updates and the activation of advanced features in your edition.

In this section you will find information regarding the different licensing options you have that will help you choose the type of order you need to place. Also this section explains how to place your order and finally activate your product so that you can enjoy all of the z/Scope benefits.

- [Licensing Information](#)
- [How to Place an Order](#)
- [Registering z/Scope Trial Version](#)

11.1 Licensing Information

When it comes to purchasing z/Scope Terminal Emulator, there are different editions

available. Our wide range of possibilities assures you that you will make the best deal.

- z/Scope Classic is Cybele Software's stand-alone, fully equipped and highly reliable terminal emulator.
- z/Scope Express 3270, z/Scope Express 5250 and z/Scope Express VT are specific, stand-alone versions, suitable for accessing IBM Mainframes (3270), IBM AS/400 (5250) and Unix (VTxxx).
- z/Scope Web-to-Host is Cybele Software's fully equipped browser based edition. It runs through Internet Explorer, Mozilla Firefox and Netscape. It includes Web Update so all users will always be updated with the version installed on the server, and it can be centrally customized according to groups' specific needs.
- z/Scope Workbench is an essential set of integrated tools specially created for Systems Administrators, Programmers and IT professionals for accessing IBM Mainframes, IBM AS/400 and Unix servers.
- z/Scope SDK is an easy-to-use library that allows you to integrate and take advantage of z/Scope Terminal Emulator within your corporate application.

To see a comparison chart visit <http://www.cybelesoft.com/en/terminal-emulator-comparison.html>

Cybele Software offers perpetual licensing for all our products. The pricing will vary according to the z/Scope edition that you choose, the optional adding of an FTP client and the amount of licenses.

Once that you have chosen the z/Scope edition that best suits your needs, if possible you can add the FTP advanced feature at a very low additional cost.

We offer Technical Support by e-mail and/or phone, which also includes free updates and upgrades during the covered period and our full commitment to timely fix bugs and problems.

We encourage users to renew the annual maintenance contract in order to be eligible for technical support and product upgrades. The maintenance fee after the first year will still be 15% of the updated price of the purchased product.

Cybele Software offers volume pricing according to the amount of the purchase.

If you have any other question, contact us at sales@cybelesoft.com. Our sales representatives will get in touch with you to assist you with your licensing situation.

11.2 Registering z/Scope Trial Version

If you downloaded z/Scope's Trial version from our web site or a distribution site and you have already purchased a license, you must follow these steps in order to register the product:

Stand-Alone Editions

Registering z/Scope Express, Classic or Workbench:

1. Go to the 'Help' menu, select the 'About' option and then press the 'Enter Key' button.
2. You will be presented with the 'Enter Key' dialog, in which you must type (you can just copy&paste) the registered User Name and Serial Number that you received when you acquired your license.



3. If you enter the Key correctly, z/Scope will no longer be operating in trial mode: your copy of the product will be now fully registered. You can verify this by checking the 'About' dialog.

Remote Configuration Server

To register your z/Scope Web-to-Host and large deployments running the Remote Configuration Server go to:

Start / Programs / zScope / Remote Configuration, then select Options > Registration. For more information about the RCS visit this online Help: <http://www.cybelesoft.com/Helps/RCM/>

z/Scope SDK

To learn how to embed your License information within your application, visit this online Help: <http://www.cybelesoft.com/Helps/zSDK/>

11.3 How to Place an Order

There are many ways to order your z/Scope licenses:

- Place an Online Order through our Web Site:

<http://www.cybelesoft.com>

- Contact us at sales@cybelesoft.com. Let us know about your licensing needs and we will send you an official quotation. Our sales representatives will get in touch with you to assist you with the purchase.
- You can also call us anytime to any of these phone numbers and place the order immediately:

Toll Free: 1-866-462-9768
Local line: 1-302-892-9625
Fax: 1-302-295-9995

- You can also contact us through Live Chat by pressing this icon in our website:



and immediately have a conversation with a representative without even having to pick up the phone.

When you buy z/Scope, you will receive a Key to register the Trial version. For instructions on how to register z/Scope when you purchase a license, see [Registering z/Scope Trial Version](#).

12 Obtaining Technical Support

Cybele's goal is to offer high quality products and services to increase the efficiency and ease-of-use of legacy systems. The whole Company focuses on this goal, and the results of our unique expertise are our reliable solutions. We believe passionately that modern, solid and feature-rich host access solutions can actually increase its users' productivity.

Technical support is a very important benefit to consider, especially when it comes to mission critical software solutions.

Using registered Cybele Software's applications not only allows you to receive free product upgrades and updates but also the certainty that you will have our team of experienced developers and technical support representatives working hard to assist you with any issue, thus making the product much more accessible in any situation.

We are here to help you out from monday to friday 9 a.m. to 5 p.m. eastern time on the phone numbers:

Toll Free: 1-866-462-9768
Local line: 1-302-892-9625
Fax: 1-302-295-9995

If you make your call outside this hour range, you can leave a message and we will get back to you.

You can send us an email to support@cybelesoft.com and we will write you back timely. You can also contact us through Live Chat by pressing this icon in our website:



and immediately have a conversation with a representative without even having to pick up the phone.



Cybele Software Inc.
3422 Old Capitol Trail, suite 1125
Wilmington, DE - 19808
Phone: (302) 892-9625
Fax: (302) 295-9995
e-mail: support@cybelesoft.com
<http://www.cybelesoft.com>