z/Scope Terminal Emulator

Administrator’s and User’s guide
# Table of Contents

## Overview

- Getting to know z/Scope  6.2  

## What’s New

- 13

## Getting to know z/Scope  6.2

### 1 The User Interface

- Start Page  
- Tab Selection Bar  
- Tiled View  

### Telnet Workspace

- Main Menu  
- Main Toolbar  
- Emulation Display  
- Printer Session Panel  
- Print Job Queue  
- Print Preview Dialog  
- Printer Parameters Dialog  
- Status Bar  

### FTP Workspace

- Main Menu  
- My Computer Panel  
- FTP Server Panel  
- File Transfer Queue  
- Queue Item Edit  
- Log Window  
- Status Bar  

### Web Browser Workspace

- Main Menu  
- Main Toolbar  
- Browser Panel  
- Status Bar  

### Key Features

- Auto-fills  
- Creating an Auto-Fill  
- Using an Auto-Fill  

### Auto-suggestion

- Using Auto-suggestion  
- Deleting Auto-suggestions  

### Favorites

- Building Your Favorites List  
- Organizing Your Favorites List  

### File Transfer

- Using the File Transfer Manager  
- FTP  
- INDYSFILE  
- KERMIT  
- X-MODEM  
- Y-MODEM  
- Z-MODEM  

© 2014, Cybele Software, Inc.
<table>
<thead>
<tr>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>3</strong></td>
</tr>
</tbody>
</table>

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
# zScope Terminal Emulator Guide v6.2

**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

© 2014, Cybele Software, Inc.
6 Navigators.......................................................................................................................................................... 219
  Editing a Navigator........................................................................................................................................... 220
    General......................................................................................................................................................... 221
    Screen Fields................................................................................................................................................ 224
    Visualization............................................................................................................................................... 225
  Deleting a Navigator....................................................................................................................................... 226

7 Auto-Fills......................................................................................................................................................... 227
  Editing an Auto-Fill......................................................................................................................................... 228
    General....................................................................................................................................................... 229
    Screen Fields............................................................................................................................................ 232
    Visualization............................................................................................................................................ 234
  Deleting an Auto-fill....................................................................................................................................... 235

8 Keyboard......................................................................................................................................................... 237
  Base............................................................................................................................................................... 238
  Host............................................................................................................................................................... 240
  Edition............................................................................................................................................................ 242
  Char............................................................................................................................................................... 244
  Custom............................................................................................................................................................ 246
  Keyboards....................................................................................................................................................... 248
    z/Scope Keyboard Map............................................................................................................................... 249
    IBM Keyboard Map................................................................................................................................. 251
    EXTRA Keyboard Map............................................................................................................................. 253
    IRMA Keyboard Map............................................................................................................................... 255
    RUMBA Keyboard Map............................................................................................................................. 257

9 Environment...................................................................................................................................................... 259
  Edit................................................................................................................................................................. 261
  Display............................................................................................................................................................ 263
  Misc................................................................................................................................................................. 265
  Plugins............................................................................................................................................................. 267
  File Transfer.................................................................................................................................................. 269
  Start Page...................................................................................................................................................... 271
  Associations................................................................................................................................................... 273
  Debug............................................................................................................................................................. 275
  Config Setup.................................................................................................................................................. 277

Programming Reference.................................................................................................................................... 279

1 Scripting Objects.................................................................................................................................................. 281
  Display Object............................................................................................................................................... 282
    Properties.................................................................................................................................................... 283
      Cols..................................................................................................................................................... 284
      Rows ................................................................................................................................................. 285
      CursorPos........................................................................................................................................ 286
      EditFields...................................................................................................................................... 288
      HostFields.................................................................................................................................... 289
      HTML............................................................................................................................................ 291
      ScreenId..................................................................................................................................... 293
      ScreenRow.................................................................................................................................... 295
      Text.................................................................................................................................................. 297
    Methods.................................................................................................................................................. 299
      Connect.......................................................................................................................................... 300
      Disconnect..................................................................................................................................... 302
      Press............................................................................................................................................... 304

© 2014, Cybele Software, Inc.
## 2 HLLAPI Reference

### 2.1 EHLLAPI Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>PressAndWait</td>
<td>273</td>
</tr>
<tr>
<td>SendAid</td>
<td>274</td>
</tr>
<tr>
<td>SendKeys</td>
<td>275</td>
</tr>
<tr>
<td>Wait</td>
<td>276</td>
</tr>
<tr>
<td>WaitFor</td>
<td>276</td>
</tr>
<tr>
<td>WaitForConnect</td>
<td>277</td>
</tr>
<tr>
<td>WaitForDisconnect</td>
<td>277</td>
</tr>
<tr>
<td>WaitForScreen</td>
<td>278</td>
</tr>
<tr>
<td>WaitForNewScreen</td>
<td>278</td>
</tr>
<tr>
<td>WaitForUnlock</td>
<td>279</td>
</tr>
<tr>
<td>WaitForNewUnlock</td>
<td>279</td>
</tr>
<tr>
<td>Cancel File Transfer</td>
<td>281</td>
</tr>
<tr>
<td>Change PS Window Name</td>
<td>282</td>
</tr>
<tr>
<td>Change Switch List LT Name</td>
<td>283</td>
</tr>
<tr>
<td>Connect Presentation Space</td>
<td>284</td>
</tr>
<tr>
<td>Connect Window Services API</td>
<td>286</td>
</tr>
<tr>
<td>Convert Position or Convert RowCol</td>
<td>287</td>
</tr>
<tr>
<td>Copy Field to String</td>
<td>289</td>
</tr>
<tr>
<td>Copy OIA (13)</td>
<td>291</td>
</tr>
<tr>
<td>Copy Presentation Space (5)</td>
<td>292</td>
</tr>
<tr>
<td>Copy Presentation Space to String (8)</td>
<td>293</td>
</tr>
<tr>
<td>Copy String to Field (33)</td>
<td>295</td>
</tr>
<tr>
<td>Copy String to Presentation Space (15)</td>
<td>296</td>
</tr>
<tr>
<td>Disconnect Presentation Space (2)</td>
<td>298</td>
</tr>
<tr>
<td>Disconnect Window Service (102)</td>
<td>299</td>
</tr>
<tr>
<td>Find Field Length</td>
<td>300</td>
</tr>
<tr>
<td>Find Field Position</td>
<td>301</td>
</tr>
<tr>
<td>Get Key (51)</td>
<td>303</td>
</tr>
<tr>
<td>Lock Presentation Space API</td>
<td>305</td>
</tr>
<tr>
<td>Lock Window Services API</td>
<td>307</td>
</tr>
<tr>
<td>Pause (18)</td>
<td>308</td>
</tr>
<tr>
<td>Post Intercept Status</td>
<td>310</td>
</tr>
<tr>
<td>Query Additional Field Attribute (45)</td>
<td>311</td>
</tr>
<tr>
<td>Query Close Intercept (42)</td>
<td>312</td>
</tr>
<tr>
<td>Query Communication Event (81)</td>
<td>312</td>
</tr>
<tr>
<td>Query Cursor Location</td>
<td>313</td>
</tr>
<tr>
<td>Query Field Attribute</td>
<td>314</td>
</tr>
<tr>
<td>Query Host Update</td>
<td>316</td>
</tr>
<tr>
<td>Query Session Status</td>
<td>317</td>
</tr>
<tr>
<td>Query Sessions (10)</td>
<td>318</td>
</tr>
<tr>
<td>Query System (20)</td>
<td>320</td>
</tr>
<tr>
<td>Query Window Coordinates (103)</td>
<td>321</td>
</tr>
<tr>
<td>Receive File (91)</td>
<td>322</td>
</tr>
<tr>
<td>Release (12)</td>
<td>325</td>
</tr>
<tr>
<td>Reserve (11)</td>
<td>325</td>
</tr>
<tr>
<td>Reset System (21)</td>
<td>326</td>
</tr>
<tr>
<td>Search Field (30)</td>
<td>327</td>
</tr>
<tr>
<td>Search Presentation Space (6)</td>
<td>329</td>
</tr>
<tr>
<td>Send File (90)</td>
<td>331</td>
</tr>
<tr>
<td>Send Key (3)</td>
<td>333</td>
</tr>
<tr>
<td>Set Cursor (40)</td>
<td>335</td>
</tr>
<tr>
<td>Set Session Parameters (9)</td>
<td>336</td>
</tr>
</tbody>
</table>
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
  WinHLLAPIFileBlocking ........................................................... 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
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.
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 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 autofills the one you would like to apply.

### Favorites menu

#### Add to Favorites
Add 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.
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.
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**

<table>
<thead>
<tr>
<th>Options</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration Settings</td>
<td>Opens the Settings</td>
</tr>
<tr>
<td>Languages</td>
<td>Allows you to choose different languages for the application.</td>
</tr>
<tr>
<td>Connection Settings</td>
<td>Opens the Connections Settings for the FTP connection.</td>
</tr>
<tr>
<td>Environment Settings</td>
<td>Opens the Environment Settings</td>
</tr>
</tbody>
</table>

**Help menu**

<table>
<thead>
<tr>
<th>Help</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Help Topics</td>
<td>Use this option to access this help.</td>
</tr>
<tr>
<td>About</td>
<td>Shows information about z/Scope.</td>
</tr>
</tbody>
</table>

**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.
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.

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
Dynamically 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 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.
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.
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

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:
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 autocompleted with the active connection, feed or web site's information.

![Add to Favorites dialog box](image)

**Name**
This 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.

   ![Add to Favorites dialog](image)

2. **Through the Favorites menu**

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

   ![Favorites menu](image)

   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](#)

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.

![File Transfer Manager Button](#)

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.

**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 predeterminated 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.
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$FILE. Then click 'Next'.
3. The Direction indicates if you are setting a download (RECEIVE) from the host to the PC, or an upload (SEND) from the PC to the host. You have the option to associate the file transfer transfer with an existing connection. You can either select it or leave it blank according to your preferences. Then click 'Next'.
If you associate the FTP File Transfer Job to an existing connection and click 'Next', the Job created will use all the settings predetermined on the connection you selected. As you do not need to enter any connectivity info, on the following screen you just need to specify the local and remote files and click 'Finish'.
Host Type
In the combobox next to the IND$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 filed you can change the name of the File Transfer program as it is defined in the host machine.

**Timeout**
Specify an 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'.

4. Choose from the combobox the specifical 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'.

Local file: C:\temp\test.txt
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'.
Please, specify the file transfer options.

- [ ] 8K
- [ ] ESC Control
- [ ] Overwrite file, if exists

**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:

**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:

**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.
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 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
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.

![Image of Macro Name window]

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.

![Welcome Wizard](image)

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.

**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 an 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
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.
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'.
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.
Additionaly 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.
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.
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'.
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 File' 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.
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.
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 an 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.
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.
settings

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

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

Related items:
- Editing an FTP Connection
- Creating a Connection
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.

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
4.1.1.1 General

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

**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:

- **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.

- 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.

- 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.

- 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.

- 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.

- 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:

**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 PRINT
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:
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:

- **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.

▲ This tab only becomes available when the 'SSL' option is checked in the 'Host' tab.
4.1.1.8 SSH

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

- **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:

**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:

**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.
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:

- **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.

  - 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:

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.

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.

- When you check this option, the 'Backup' tab will automatically appear in the
SSL
Enables the SSL (Secure Sockets Layer) protocol for the host.

- 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.

- 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.

- 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:
**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:

- **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**

  Device Name Suffix:
  - None
  - Letter (A-Z)
  - Number (1-9)
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:

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:

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:

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Indicates the type of Socks protocol you will be connecting to. z/Scope provides support for svSocks 4, 4A and 5 protocols.</td>
</tr>
<tr>
<td>Address</td>
<td>In this field you must enter the IP address of the Socks server.</td>
</tr>
<tr>
<td>Port</td>
<td>In this field you must enter the port number of the Socks service at the host.</td>
</tr>
<tr>
<td>Requires Authentication</td>
<td>When connecting with svSocks4A protocol and higher, you have the option of providing a User ID and a Password for authentication.</td>
</tr>
<tr>
<td>UserId</td>
<td>In this field you must enter your User ID.</td>
</tr>
<tr>
<td>Password</td>
<td></td>
</tr>
</tbody>
</table>

This tab only becomes available when the 'Socks' option is checked in the 'Host' tab.
In this field you must enter your Password.

4.1.2.7 SSL

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

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:

**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 want 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.

4.1.2.12 Keypads

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

- **Connection name**
  In this field you must enter a name for 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:
### 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.](image)
  - 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.](image)

### 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:
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:

**Connection name**
In this field you must enter a name for 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:

- **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:
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)

### 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:

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:

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:

**Domain\User:Password**
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:

![Field Options Tab](image)

**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:
**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:

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:

  1. 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.
  2. 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 (i.e. {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 these steps:
  1. Click on the 'Test' button. The 'Test Hotspot Rule' dialog will be displayed.

     ![Test Hotspot Rule dialog]

     Test string: ENTER
     The resulting keystrokes will be: {ENTER}

     Test Exit

     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.

   ! Warning: 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:

![Hotspot Configuration Window]

**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.

**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 technics, 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:
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 Fields Tab](image)

**Screen unique ID number**
This number is used to univocally identify the screen the Navigator applies to.

[Delete Screen] 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.

[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 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:

[Switch To Ask For Input] 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.

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.

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 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.

© 2014, Cybele Software, Inc.
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:
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 user to univocally identify the screen the Auto-Fill applies to.

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.

Click on this button to open the Active Script editor window where you will write you 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.

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.

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.
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-Fill Interface]

⚠️ 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

<table>
<thead>
<tr>
<th>Aids</th>
<th>Mapped keys</th>
</tr>
</thead>
<tbody>
<tr>
<td>PF01 .. PF12</td>
<td>F1 .. F12</td>
</tr>
</tbody>
</table>
### 5250 Keyboard

<table>
<thead>
<tr>
<th>Aids</th>
<th>Mapped keys</th>
</tr>
</thead>
<tbody>
<tr>
<td>PF01 .. PF12</td>
<td>F1 .. F12</td>
</tr>
<tr>
<td>PF13 .. PF24</td>
<td>Shift + F1 .. Shift + F12</td>
</tr>
<tr>
<td>PA01 .. PA03</td>
<td>Left Control + F1 .. Left Control + F3</td>
</tr>
<tr>
<td>Attention</td>
<td>Escape</td>
</tr>
<tr>
<td>Clear</td>
<td>Pause</td>
</tr>
<tr>
<td></td>
<td>Shift + Pause</td>
</tr>
<tr>
<td></td>
<td>Left Ctrl + Shift + Z</td>
</tr>
<tr>
<td>Enter</td>
<td>Enter</td>
</tr>
<tr>
<td></td>
<td>Shift + Enter</td>
</tr>
<tr>
<td>Erase Input</td>
<td>Left Alt + End</td>
</tr>
<tr>
<td>Reset</td>
<td>Left Control</td>
</tr>
<tr>
<td></td>
<td>Left Control + R</td>
</tr>
<tr>
<td>System Request</td>
<td>Left Alt + PrintScreen</td>
</tr>
<tr>
<td></td>
<td>Left Alt + Multiply</td>
</tr>
<tr>
<td>Test Request</td>
<td>Left Alt + Pause</td>
</tr>
</tbody>
</table>

If your keyboard does not distinguish keys then left keys are equivalent to right keys.
### IBM Keyboard Map

#### IBM 3270 Keyboard Map

<table>
<thead>
<tr>
<th>Function Key</th>
<th>Mapped keys</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attn</td>
<td>Esc</td>
</tr>
<tr>
<td>Back Space</td>
<td>Backspace</td>
</tr>
<tr>
<td>Back Tab</td>
<td>Left Shift + Tab</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Tab</td>
</tr>
<tr>
<td>Clear</td>
<td>Pause</td>
</tr>
<tr>
<td>Cursor Down</td>
<td>Down</td>
</tr>
<tr>
<td></td>
<td>Num 2</td>
</tr>
<tr>
<td>Cursor Down &amp; Select</td>
<td>Left Shift + Down</td>
</tr>
<tr>
<td></td>
<td>Left Shift + Num 2</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Down</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Num 2</td>
</tr>
<tr>
<td>Cursor Left</td>
<td>Left</td>
</tr>
<tr>
<td></td>
<td>Num 4</td>
</tr>
<tr>
<td>Cursor Left &amp; Select</td>
<td>Left Shift + Left</td>
</tr>
<tr>
<td></td>
<td>Left Shift + Num 4</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Left</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Num 4</td>
</tr>
<tr>
<td>Cursor Right</td>
<td>Right</td>
</tr>
<tr>
<td></td>
<td>Num 6</td>
</tr>
<tr>
<td>Cursor Right &amp; Select</td>
<td>Left Shift + Right</td>
</tr>
<tr>
<td></td>
<td>Left Shift + Num 6</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Right</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Num 6</td>
</tr>
<tr>
<td>Cursor Ruler</td>
<td>Left Control + Home</td>
</tr>
<tr>
<td></td>
<td>Right Control + Home</td>
</tr>
<tr>
<td>Cursor Up</td>
<td>Up</td>
</tr>
<tr>
<td></td>
<td>Num 8</td>
</tr>
<tr>
<td>Cursor Up &amp; Select</td>
<td>Left Shift + Up</td>
</tr>
<tr>
<td></td>
<td>Left Shift + Num 8</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Up</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Num 8</td>
</tr>
<tr>
<td>Delete</td>
<td>Delete</td>
</tr>
<tr>
<td></td>
<td>Num Del</td>
</tr>
<tr>
<td>Dup</td>
<td>Left Shift + Insert</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Insert</td>
</tr>
<tr>
<td>End</td>
<td>End</td>
</tr>
<tr>
<td></td>
<td>Num 1</td>
</tr>
<tr>
<td>Enter</td>
<td>Num Enter</td>
</tr>
<tr>
<td></td>
<td>Right Control + Right Ctrl</td>
</tr>
<tr>
<td>Erase Eof</td>
<td>Left Control + End</td>
</tr>
<tr>
<td></td>
<td>Right Control + End</td>
</tr>
<tr>
<td>Erase Input</td>
<td>Left Alt + End</td>
</tr>
<tr>
<td></td>
<td>Right Alt + End</td>
</tr>
<tr>
<td>Field Mark</td>
<td>Left Shift + Home</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Home</td>
</tr>
<tr>
<td>Function Key</td>
<td>Mapped keys</td>
</tr>
<tr>
<td>-------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>Attn</td>
<td>Esc</td>
</tr>
<tr>
<td>Back Space</td>
<td>Backspace</td>
</tr>
<tr>
<td>Back Tab</td>
<td>Left Shift + Tab</td>
</tr>
<tr>
<td></td>
<td>Left Shift + Tab</td>
</tr>
<tr>
<td>Cursor Down</td>
<td>Down</td>
</tr>
<tr>
<td></td>
<td>Num 2</td>
</tr>
<tr>
<td>Cursor Down &amp; Select</td>
<td>Left Shift + Down</td>
</tr>
<tr>
<td></td>
<td>Left Shift + Num 2</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Down</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Num 2</td>
</tr>
<tr>
<td>Cursor Left</td>
<td>Left</td>
</tr>
<tr>
<td></td>
<td>Num 4</td>
</tr>
<tr>
<td>Cursor Left &amp; Select</td>
<td>Left Shift + Left</td>
</tr>
<tr>
<td></td>
<td>Left Shift + Num 4</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Left</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Num 4</td>
</tr>
<tr>
<td>Cursor Right</td>
<td>Right</td>
</tr>
<tr>
<td></td>
<td>Num 6</td>
</tr>
<tr>
<td>Cursor Right &amp; Select</td>
<td>Left Shift + Right</td>
</tr>
<tr>
<td></td>
<td>Left Shift + Num 6</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Right</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Num 6</td>
</tr>
<tr>
<td>Cursor Ruler</td>
<td>Left Control + Home</td>
</tr>
<tr>
<td></td>
<td>Right Control + Home</td>
</tr>
<tr>
<td>Cursor Up</td>
<td>Up</td>
</tr>
<tr>
<td></td>
<td>Num 8</td>
</tr>
<tr>
<td>Cursor Up &amp; Select</td>
<td>Left Shift + Up</td>
</tr>
<tr>
<td></td>
<td>Left Shift + Num 8</td>
</tr>
</tbody>
</table>

If your keyboard does not distinguish keys then left keys are equivalent to right keys.

**IBM 5250 Keyboard Map**

If your keyboard does not distinguish keys then left keys are equivalent to right keys.
<table>
<thead>
<tr>
<th>Function Key</th>
<th>Mapped keys</th>
</tr>
</thead>
<tbody>
<tr>
<td>Right Shift + Up</td>
<td>Num Del</td>
</tr>
<tr>
<td>Right Shift + Num 8</td>
<td></td>
</tr>
<tr>
<td>Delete</td>
<td>Delete</td>
</tr>
<tr>
<td></td>
<td>Num Del</td>
</tr>
<tr>
<td>Dup</td>
<td>Left Shift + Insert</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Insert</td>
</tr>
<tr>
<td>End</td>
<td>End</td>
</tr>
<tr>
<td>Enter</td>
<td>Right Control + Right Ctrl</td>
</tr>
<tr>
<td>Erase Eof</td>
<td>Right Control + End</td>
</tr>
<tr>
<td></td>
<td>Left Control + End</td>
</tr>
<tr>
<td>Erase Input</td>
<td>Left Alt + End</td>
</tr>
<tr>
<td></td>
<td>Right Alt + End</td>
</tr>
<tr>
<td>Field -</td>
<td>Num -</td>
</tr>
<tr>
<td></td>
<td>Left Shift + Num -</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Num -</td>
</tr>
<tr>
<td></td>
<td>Left Alt + Num -</td>
</tr>
<tr>
<td></td>
<td>Right Alt + Num -</td>
</tr>
<tr>
<td>Field +</td>
<td>Num Enter</td>
</tr>
<tr>
<td></td>
<td>Left Shift + Num Enter</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Num Enter</td>
</tr>
<tr>
<td></td>
<td>Num +</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Num +</td>
</tr>
<tr>
<td></td>
<td>Left Shift + Num +</td>
</tr>
<tr>
<td>Field Mark</td>
<td>Left Shift + Home</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Home</td>
</tr>
<tr>
<td>Help</td>
<td>Left Alt + F1</td>
</tr>
<tr>
<td></td>
<td>Right Alt + F1</td>
</tr>
<tr>
<td>Home</td>
<td>Home</td>
</tr>
<tr>
<td></td>
<td>Num 7</td>
</tr>
<tr>
<td>Home</td>
<td>Home</td>
</tr>
<tr>
<td></td>
<td>Num 7</td>
</tr>
<tr>
<td>Insert</td>
<td>Insert</td>
</tr>
<tr>
<td></td>
<td>Num 0</td>
</tr>
<tr>
<td>New Line</td>
<td>Enter</td>
</tr>
<tr>
<td>PA1</td>
<td>Left Alt + Insert</td>
</tr>
<tr>
<td></td>
<td>Right Alt + Insert</td>
</tr>
<tr>
<td>PA2</td>
<td>Left Alt + Home</td>
</tr>
<tr>
<td></td>
<td>Right Alt + Home</td>
</tr>
<tr>
<td>PA3</td>
<td>Left Shift + Page Up</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Page Up</td>
</tr>
</tbody>
</table>

- 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

<table>
<thead>
<tr>
<th>Function Key</th>
<th>Mapped keys</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key</td>
<td>Shortcuts</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>Attn</td>
<td>Left Shift + Esc&lt;br&gt;Right Shift + Esc</td>
</tr>
<tr>
<td>Back Space</td>
<td>Backspace</td>
</tr>
<tr>
<td>Back Tab</td>
<td>Left Shift + Tab&lt;br&gt;Right Shift + Tab</td>
</tr>
<tr>
<td>Clear</td>
<td>Pause</td>
</tr>
<tr>
<td>Cursor Down</td>
<td>Down&lt;br&gt;Num 2</td>
</tr>
<tr>
<td>Cursor Down &amp; Select</td>
<td>Left Shift + Down&lt;br&gt;Left Shift + Num 2&lt;br&gt;Right Shift + Down&lt;br&gt;Right Shift + Num 2</td>
</tr>
<tr>
<td>Cursor Left</td>
<td>Left&lt;br&gt;Num 4</td>
</tr>
<tr>
<td>Cursor Left &amp; Select</td>
<td>Left Shift + Left&lt;br&gt;Left Shift + Num 4&lt;br&gt;Right Shift + Left&lt;br&gt;Right Shift + Num 4</td>
</tr>
<tr>
<td>Cursor Right</td>
<td>Right&lt;br&gt;Num 6</td>
</tr>
<tr>
<td>Cursor Right &amp; Select</td>
<td>Left Shift + Right&lt;br&gt;Left Shift + Num 6&lt;br&gt;Right Shift + Right&lt;br&gt;Right Shift + Num 6</td>
</tr>
<tr>
<td>Cursor Ruler</td>
<td>Left Control + Num 7&lt;br&gt;Right Control + Num 7</td>
</tr>
<tr>
<td>Cursor Up</td>
<td>Up&lt;br&gt;Num 8</td>
</tr>
<tr>
<td>Cursor Up &amp; Select</td>
<td>Left Shift + Up&lt;br&gt;Left Shift + Num 8&lt;br&gt;Right Shift + Up&lt;br&gt;Right Shift + Num 8</td>
</tr>
<tr>
<td>Delete</td>
<td>Delete&lt;br&gt;Num Del</td>
</tr>
<tr>
<td>Dup</td>
<td>Left Shift + Insert&lt;br&gt;Right Shift + Insert</td>
</tr>
<tr>
<td>End</td>
<td>End</td>
</tr>
<tr>
<td>Enter</td>
<td>Num Enter&lt;br&gt;Enter</td>
</tr>
<tr>
<td>Erase Eof</td>
<td>Num 1&lt;br&gt;End</td>
</tr>
<tr>
<td>Erase Input</td>
<td>Left Alt + End&lt;br&gt;Right Alt + End</td>
</tr>
<tr>
<td>Field Mark</td>
<td>Left Shift + Home&lt;br&gt;Right Shift + Home</td>
</tr>
<tr>
<td>Home</td>
<td>Home&lt;br&gt;Num 7</td>
</tr>
<tr>
<td>Insert</td>
<td>Insert&lt;br&gt;Num 0</td>
</tr>
<tr>
<td>New Line</td>
<td>Right Control + Right Ctrl</td>
</tr>
</tbody>
</table>

© 2014, Cybele Software, Inc.
If your keyboard does not distinguish keys then left keys are equivalent to right keys.

**EXTRA 5250 Keyboard Map**

<table>
<thead>
<tr>
<th>Function Key</th>
<th>Mapped keys</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attn</td>
<td>Left Shift + Esc</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Esc</td>
</tr>
<tr>
<td>Back Space</td>
<td>Backspace</td>
</tr>
<tr>
<td>Back Tab</td>
<td>Right Shift + Tab</td>
</tr>
<tr>
<td></td>
<td>Left Shift + Tab</td>
</tr>
<tr>
<td>Cursor Down</td>
<td>Down</td>
</tr>
<tr>
<td></td>
<td>Num 2</td>
</tr>
<tr>
<td>Cursor Down &amp; Select</td>
<td>Left Shift + Down</td>
</tr>
<tr>
<td></td>
<td>Left Shift + Num 2</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Down</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Num 2</td>
</tr>
<tr>
<td>Cursor Left</td>
<td>Left</td>
</tr>
<tr>
<td></td>
<td>Num 4</td>
</tr>
<tr>
<td>Cursor Left &amp; Select</td>
<td>Left Shift + Left</td>
</tr>
<tr>
<td></td>
<td>Left Shift + Num 4</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Left</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Num 4</td>
</tr>
<tr>
<td>Cursor Right</td>
<td>Right</td>
</tr>
<tr>
<td></td>
<td>Num 6</td>
</tr>
<tr>
<td>Cursor Right &amp; Select</td>
<td>Left Shift + Right</td>
</tr>
<tr>
<td></td>
<td>Left Shift + Num 6</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Right</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Num 6</td>
</tr>
<tr>
<td>Cursor Ruler</td>
<td>Left Control + Num 7</td>
</tr>
<tr>
<td></td>
<td>Right Control + Num 7</td>
</tr>
<tr>
<td>Cursor Up</td>
<td>Up</td>
</tr>
<tr>
<td></td>
<td>Num 8</td>
</tr>
<tr>
<td>Cursor Up &amp; Select</td>
<td>Left Shift + Up</td>
</tr>
<tr>
<td></td>
<td>Left Shift + Num 8</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Up</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Num 8</td>
</tr>
<tr>
<td>Delete</td>
<td>Delete</td>
</tr>
<tr>
<td></td>
<td>Num Del</td>
</tr>
<tr>
<td>Dup</td>
<td>Left Shift + Insert</td>
</tr>
<tr>
<td>Function Key</td>
<td>Mapped Keys</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Attn</td>
<td>F1</td>
</tr>
<tr>
<td>Back Space</td>
<td>Backspace</td>
</tr>
<tr>
<td>Back Tab</td>
<td>Left Shift + Tab</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Tab</td>
</tr>
<tr>
<td>Clear</td>
<td>F2</td>
</tr>
<tr>
<td>Cursor Down</td>
<td>Down</td>
</tr>
<tr>
<td></td>
<td>Num 2</td>
</tr>
<tr>
<td>Cursor Down &amp; Select</td>
<td>Left Shift + Down</td>
</tr>
<tr>
<td>Function</td>
<td>Key Combinations</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td>Cursor Left</td>
<td>Left Shift + Num 2, Right Shift + Down</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Num 2</td>
</tr>
<tr>
<td>Cursor Right</td>
<td>Right Shift + Num 6</td>
</tr>
<tr>
<td>Cursor Left &amp; Select</td>
<td>Left Shift + Left, Right Shift + Num 4</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Left, Right Shift + Num 4</td>
</tr>
<tr>
<td>Cursor Right &amp; Select</td>
<td>Left Shift + Right, Right Shift + Num 6</td>
</tr>
<tr>
<td>Cursor Ruler</td>
<td>Left Control + Home, Right Control + Home</td>
</tr>
<tr>
<td>Cursor Up</td>
<td>Up, Num 8</td>
</tr>
<tr>
<td>Cursor Up &amp; Select</td>
<td>Left Shift + Up, Right Shift + Num 8</td>
</tr>
<tr>
<td>Delete</td>
<td>Delete, Num Del</td>
</tr>
<tr>
<td>Dup</td>
<td>Left Control + G, Right Control + G</td>
</tr>
<tr>
<td>End</td>
<td>End, Num 1</td>
</tr>
<tr>
<td>Enter</td>
<td>Enter, Num Enter</td>
</tr>
<tr>
<td>Erase Eof</td>
<td>F6</td>
</tr>
<tr>
<td>Erase Input</td>
<td>F4</td>
</tr>
<tr>
<td>Field Mark</td>
<td>Left Control + H, Right Control + H</td>
</tr>
<tr>
<td>Home</td>
<td>Home, Num 7</td>
</tr>
<tr>
<td>Insert</td>
<td>Num 0, Insert</td>
</tr>
<tr>
<td>New Line</td>
<td>Left Control + Enter, Right Control + Enter</td>
</tr>
<tr>
<td>PA1</td>
<td>Left Control + J, Right Control + J</td>
</tr>
<tr>
<td>PA2</td>
<td>Left Control + K, Right Control + K</td>
</tr>
<tr>
<td>PA3</td>
<td>Left Control + L, Right Control + L</td>
</tr>
<tr>
<td>PF01</td>
<td>Left Alt + 1, Right Alt + 1</td>
</tr>
<tr>
<td>PF02</td>
<td>Left Alt + 2</td>
</tr>
</tbody>
</table>
If your keyboard does not distinguish keys then left keys are equivalent to right keys.

**IRMA 5250 Keyboard Map**

<table>
<thead>
<tr>
<th>Function Key</th>
<th>Mapped keys</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attn</td>
<td>Esc</td>
</tr>
<tr>
<td>Back Space</td>
<td>Backspace</td>
</tr>
<tr>
<td>Back Tab</td>
<td>Right Shift + Tab</td>
</tr>
<tr>
<td></td>
<td>Left Shift + Tab</td>
</tr>
<tr>
<td>Cursor Down</td>
<td>Down</td>
</tr>
<tr>
<td></td>
<td>Num 2</td>
</tr>
<tr>
<td>Cursor Down &amp; Select</td>
<td>Left Shift + Down</td>
</tr>
<tr>
<td></td>
<td>Left Shift + Num 2</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Down</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Num 2</td>
</tr>
<tr>
<td>Cursor Left</td>
<td>Left</td>
</tr>
<tr>
<td></td>
<td>Num 4</td>
</tr>
<tr>
<td>Cursor Left &amp; Select</td>
<td>Left Shift + Left</td>
</tr>
<tr>
<td></td>
<td>Left Shift + Num 4</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Left</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Num 4</td>
</tr>
<tr>
<td>Cursor Right</td>
<td>Right</td>
</tr>
<tr>
<td></td>
<td>Num 6</td>
</tr>
<tr>
<td>Cursor Right &amp; Select</td>
<td>Left Shift + Right</td>
</tr>
<tr>
<td></td>
<td>Left Shift + Num 6</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Right</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Num 6</td>
</tr>
<tr>
<td>Cursor Ruler</td>
<td>Left Control + Home</td>
</tr>
<tr>
<td></td>
<td>Right Control + Home</td>
</tr>
<tr>
<td>Cursor Up</td>
<td>Up</td>
</tr>
<tr>
<td></td>
<td>Num 8</td>
</tr>
<tr>
<td>Cursor Up &amp; Select</td>
<td>Left Shift + Up</td>
</tr>
<tr>
<td></td>
<td>Left Shift + Num 8</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Up</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Num 8</td>
</tr>
<tr>
<td>Delete</td>
<td>Delete</td>
</tr>
<tr>
<td></td>
<td>Num Del</td>
</tr>
<tr>
<td>Dup</td>
<td>Left Shift + Right Control + Insert</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Insert</td>
</tr>
<tr>
<td>End</td>
<td>End</td>
</tr>
<tr>
<td>Enter</td>
<td>Enter</td>
</tr>
<tr>
<td>Erase Eof</td>
<td>Left Control + End</td>
</tr>
<tr>
<td>Erase Input</td>
<td>Left Alt + End</td>
</tr>
<tr>
<td>Field -</td>
<td>Num -</td>
</tr>
</tbody>
</table>
| 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 | 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

<table>
<thead>
<tr>
<th>Function Key</th>
<th>Mapped keys</th>
</tr>
</thead>
</table>
| 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  

© 2014, Cybele Software, Inc.
<table>
<thead>
<tr>
<th>Command</th>
<th>Keys/Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Num 2</td>
<td></td>
</tr>
<tr>
<td>Cursor Down &amp; Select</td>
<td>Left Shift + Down</td>
</tr>
<tr>
<td></td>
<td>Left Shift + Num 2</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Down</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Num 2</td>
</tr>
<tr>
<td>Cursor Left</td>
<td>Left</td>
</tr>
<tr>
<td></td>
<td>Num 4</td>
</tr>
<tr>
<td>Cursor Left &amp; Select</td>
<td>Left Shift + Left</td>
</tr>
<tr>
<td></td>
<td>Left Shift + Num 4</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Left</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Num 4</td>
</tr>
<tr>
<td>Cursor Right</td>
<td>Right</td>
</tr>
<tr>
<td></td>
<td>Num 6</td>
</tr>
<tr>
<td>Cursor Right &amp; Select</td>
<td>Left Shift + Right</td>
</tr>
<tr>
<td></td>
<td>Left Shift + Num 6</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Right</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Num 6</td>
</tr>
<tr>
<td>Cursor Ruler</td>
<td>Left Control + Home</td>
</tr>
<tr>
<td>Cursor Up</td>
<td>Up</td>
</tr>
<tr>
<td></td>
<td>Num 8</td>
</tr>
<tr>
<td>Cursor Up &amp; Select</td>
<td>Left Shift + Up</td>
</tr>
<tr>
<td></td>
<td>Left Shift + Num 8</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Up</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Num 8</td>
</tr>
<tr>
<td>Delete</td>
<td>Delete</td>
</tr>
<tr>
<td></td>
<td>Num Del</td>
</tr>
<tr>
<td>Dup</td>
<td>Left Control + D</td>
</tr>
<tr>
<td></td>
<td>Right Control + D</td>
</tr>
<tr>
<td>End</td>
<td>End</td>
</tr>
<tr>
<td>Enter</td>
<td>Enter</td>
</tr>
<tr>
<td></td>
<td>Num Enter</td>
</tr>
<tr>
<td>Erase Eof</td>
<td>End</td>
</tr>
<tr>
<td>Erase Input</td>
<td>Left Control + T</td>
</tr>
<tr>
<td></td>
<td>Right Control + T</td>
</tr>
<tr>
<td>Field Mark</td>
<td>Left Shift + Home</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Home</td>
</tr>
<tr>
<td>Home</td>
<td>Home</td>
</tr>
<tr>
<td></td>
<td>Num 7</td>
</tr>
<tr>
<td>Insert</td>
<td>Insert</td>
</tr>
<tr>
<td></td>
<td>Num 0</td>
</tr>
<tr>
<td>New Line</td>
<td>Left Control + Enter</td>
</tr>
<tr>
<td>PA1</td>
<td>Left Control + F1</td>
</tr>
<tr>
<td></td>
<td>Right Control + F1</td>
</tr>
<tr>
<td>PA2</td>
<td>Left Control + F2</td>
</tr>
<tr>
<td></td>
<td>Right Control + F2</td>
</tr>
<tr>
<td>PA3</td>
<td>Left Control + F3</td>
</tr>
<tr>
<td></td>
<td>Right Control + F3</td>
</tr>
<tr>
<td>PF01</td>
<td>F1</td>
</tr>
<tr>
<td>PF02</td>
<td>F2</td>
</tr>
<tr>
<td>PF03</td>
<td>F3</td>
</tr>
</tbody>
</table>
If your keyboard does not distinguish keys then left keys are equivalent to right keys.

### RUMBA 5250 Keyboard Map

<table>
<thead>
<tr>
<th>Function Key</th>
<th>Mapped keys</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attn</td>
<td>Esc</td>
</tr>
<tr>
<td>Back Space</td>
<td>Backspace</td>
</tr>
<tr>
<td></td>
<td>Left Shift + Backspace</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Backspace</td>
</tr>
<tr>
<td>Back Tab</td>
<td>Right Shift + Tab</td>
</tr>
<tr>
<td></td>
<td>Left Shift + Tab</td>
</tr>
<tr>
<td>Cursor Down</td>
<td>Down</td>
</tr>
<tr>
<td></td>
<td>Num 2</td>
</tr>
<tr>
<td>Cursor Down &amp; Select</td>
<td>Left Shift + Down</td>
</tr>
<tr>
<td></td>
<td>Left Shift + Num 2</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Down</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Num 2</td>
</tr>
<tr>
<td>Cursor Left</td>
<td>Left</td>
</tr>
<tr>
<td></td>
<td>Num 4</td>
</tr>
<tr>
<td>Cursor Left &amp; Select</td>
<td>Left Shift + Left</td>
</tr>
<tr>
<td></td>
<td>Left Shift + Num 4</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Left</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Num 4</td>
</tr>
<tr>
<td>Cursor Right</td>
<td>Right</td>
</tr>
<tr>
<td></td>
<td>Num 6</td>
</tr>
<tr>
<td>Cursor Right &amp; Select</td>
<td>Left Shift + Right</td>
</tr>
<tr>
<td></td>
<td>Left Shift + Num 6</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Right</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Num 6</td>
</tr>
<tr>
<td>Cursor Ruler</td>
<td>Left Alt + Page Down</td>
</tr>
<tr>
<td></td>
<td>Right Alt + Page Down</td>
</tr>
<tr>
<td>Cursor Up</td>
<td>Up</td>
</tr>
<tr>
<td></td>
<td>Num 8</td>
</tr>
<tr>
<td>Cursor Up &amp; Select</td>
<td>Left Shift + Up</td>
</tr>
<tr>
<td></td>
<td>Left Shift + Num 8</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Up</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Num 8</td>
</tr>
<tr>
<td>Delete</td>
<td>Delete</td>
</tr>
<tr>
<td></td>
<td>Num Del</td>
</tr>
<tr>
<td>Dup</td>
<td>Left Shift + Insert</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Insert</td>
</tr>
<tr>
<td>End</td>
<td>End</td>
</tr>
<tr>
<td>Enter</td>
<td>Enter</td>
</tr>
<tr>
<td>Erase Eof</td>
<td>Left Control + End</td>
</tr>
<tr>
<td></td>
<td>Right Control + End</td>
</tr>
<tr>
<td>Erase Input</td>
<td>Left Alt + End</td>
</tr>
<tr>
<td></td>
<td>Right Alt + End</td>
</tr>
<tr>
<td>Field -</td>
<td>Num -</td>
</tr>
<tr>
<td>Function</td>
<td>Key combinations</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>Left Shift + Num -</td>
<td>Right Shift + Num -</td>
</tr>
<tr>
<td>Field +</td>
<td>Right Control + Right Ctrl Num +</td>
</tr>
<tr>
<td>Field Mark</td>
<td>Left Shift + Home</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Home</td>
</tr>
<tr>
<td>Help</td>
<td>Scroll Lock</td>
</tr>
<tr>
<td>Home</td>
<td>Home</td>
</tr>
<tr>
<td></td>
<td>Num 7</td>
</tr>
<tr>
<td>Insert</td>
<td>Insert</td>
</tr>
<tr>
<td></td>
<td>Num 0</td>
</tr>
<tr>
<td>New Line</td>
<td>Left Shift + Enter</td>
</tr>
<tr>
<td></td>
<td>Right Shift + Enter</td>
</tr>
<tr>
<td>PA1</td>
<td>Left Control + F1</td>
</tr>
<tr>
<td></td>
<td>Right Control + F1</td>
</tr>
<tr>
<td>PA2</td>
<td>Left Control + F2</td>
</tr>
<tr>
<td></td>
<td>Right Control + F2</td>
</tr>
<tr>
<td>PA3</td>
<td>Left Control + F3</td>
</tr>
<tr>
<td></td>
<td>Right Control + F3</td>
</tr>
</tbody>
</table>

⚠️ **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:
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:

![Associations Tab](image)

**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:

**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](#)
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

\[ \text{Display}.\text{Cols} \]

Remarks

Possible values for this property include for example 80 and 132, depending on the terminal type you are working with.

Related Topics

- \text{Rows Property}

5.1.1.1.2 Rows

The Rows property retrieves the total number of rows supported by the current terminal type.

Syntax

\[ \text{Display}.\text{Rows} \]

Remarks

Possible values for this property include for example 24, 32 and 43, depending on the terminal type you are working with.

Related Topics

- \text{Cols Property}

5.1.1.1.3 CursorPos

The CursorPos property retrieves and sets the cursor's current screen position.

Syntax

\[ \text{Display}.\text{CursorPos} [ = \text{cursor position}] \]
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:

\[
\text{Cursor position} = \text{Total columns} \times (\text{cursor's row number} - 1) + \text{cursor's column number}
\]

5.1.1.4 EditFields

EditFields is a collection that contains all the editable (unprotected) fields in the current screen.

Syntax

\[
[\text{string} = \text{Display}.\text{EditFields}(\text{index|name}) = [\text{string}]\]
\]

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

5.1.1.5 HostFields

HostFields is a collection that contains all the fields in the current screen.
Syntax

\[ \text{string} = \] Display\texttt{.HostFields(\text{index}|\text{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[\text{row}]C[\text{column}]. For example, the field at row 4 column 30 in the screen can be referenced as "R4C30".

\text{z}/\text{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

- \texttt{EditFields Collection}

5.1.1.1.6 HTML

The HTML property returns the current emulation in an HTML page for each screen.

Syntax

\[ \text{string} = \] Display\texttt{.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

\[ \text{Integer} = \] Display\texttt{.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

\[
\text{Display}.\text{ScreenRow}(\text{Row As Integer})
\]

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

\[
\text{Display}.\text{Text}([\text{row}], [\text{col}], [\text{len}])
\]

or

\[
\text{Display}.\text{Text}(\text{R[row]C[col]L[len]})
\]
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

\[
[\text{boolean}] = \text{Display}.\text{Connect}([\text{timeout}])
\]

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.
5.1.1.2.2 Disconnect

The Disconnect method terminates the connection to the Telnet server.

Syntax

\[
\text{[boolean]} = \text{Display.}\text{Disconnect([timeout])}
\]

Remarks

The \textit{timeout} parameter is optional and is measured in milliseconds. This method returns 'True' if the connection is successfully established and 'False' if the \textit{timeout} period expires.

If you specify a value in the \textit{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.\textit{Press}(\textit{AidKey} \text{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.\textit{PressAndWait}(\textit{AidKey} \text{As String})

Related Topics
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:

<table>
<thead>
<tr>
<th>String Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enter</td>
<td>Enter key</td>
</tr>
<tr>
<td>PF1</td>
<td>PF1 Key</td>
</tr>
<tr>
<td>PF2</td>
<td>PF2 Key</td>
</tr>
<tr>
<td>PF3</td>
<td>PF3 Key</td>
</tr>
<tr>
<td>PF4</td>
<td>PF4 Key</td>
</tr>
<tr>
<td>PF5</td>
<td>PF5 Key</td>
</tr>
<tr>
<td>PF6</td>
<td>PF5 Key</td>
</tr>
<tr>
<td>PF7</td>
<td>PF7 Key</td>
</tr>
<tr>
<td>PF8</td>
<td>PF8 Key</td>
</tr>
<tr>
<td>PF9</td>
<td>PF9 Key</td>
</tr>
<tr>
<td>PF10</td>
<td>PF10 Key</td>
</tr>
<tr>
<td>PF11</td>
<td>PF11 Key</td>
</tr>
<tr>
<td>PF12</td>
<td>PF12 Key</td>
</tr>
<tr>
<td>PF13</td>
<td>PF13 Key</td>
</tr>
<tr>
<td>PF14</td>
<td>PF14 Key</td>
</tr>
<tr>
<td>PF15</td>
<td>PF15 Key</td>
</tr>
<tr>
<td>PF16</td>
<td>PF16 Key</td>
</tr>
<tr>
<td>PF17</td>
<td>PF17 Key</td>
</tr>
<tr>
<td>PF18</td>
<td>PF18 Key</td>
</tr>
<tr>
<td>PF19</td>
<td>PF19 Key</td>
</tr>
<tr>
<td>PF20</td>
<td>PF20 Key</td>
</tr>
<tr>
<td>PF21</td>
<td>PF21 Key</td>
</tr>
<tr>
<td>PF22</td>
<td>PF22 Key</td>
</tr>
<tr>
<td>PF23</td>
<td>PF23 Key</td>
</tr>
<tr>
<td>PF24</td>
<td>PF24 Key</td>
</tr>
<tr>
<td>PA1</td>
<td>PA1 Key</td>
</tr>
<tr>
<td>PA2</td>
<td>PA2 Key</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>@A@Q</td>
<td>Attention</td>
</tr>
<tr>
<td>@&lt;</td>
<td>Backspace</td>
</tr>
<tr>
<td>@B</td>
<td>BackTab (Left Tab)</td>
</tr>
<tr>
<td>@C</td>
<td>Clear</td>
</tr>
<tr>
<td>@E</td>
<td>Enter</td>
</tr>
<tr>
<td>@F</td>
<td>Erase Field</td>
</tr>
<tr>
<td>@A@Q</td>
<td>Sys Request</td>
</tr>
<tr>
<td>@T</td>
<td>Tab (Right Tab)</td>
</tr>
<tr>
<td>@x</td>
<td>PA1</td>
</tr>
<tr>
<td>@y</td>
<td>PA2</td>
</tr>
<tr>
<td>@z</td>
<td>PA3</td>
</tr>
<tr>
<td>@1</td>
<td>PF1</td>
</tr>
<tr>
<td>@2</td>
<td>PF2</td>
</tr>
<tr>
<td>@3</td>
<td>PF3</td>
</tr>
<tr>
<td>@4</td>
<td>PF4</td>
</tr>
<tr>
<td>@5</td>
<td>PF5</td>
</tr>
<tr>
<td>@6</td>
<td>PF6</td>
</tr>
<tr>
<td>@7</td>
<td>PF7</td>
</tr>
<tr>
<td>@8</td>
<td>PF8</td>
</tr>
<tr>
<td>@9</td>
<td>PF9</td>
</tr>
<tr>
<td>@a</td>
<td>PF10</td>
</tr>
<tr>
<td>@b</td>
<td>PF11</td>
</tr>
<tr>
<td>@c</td>
<td>PF12</td>
</tr>
<tr>
<td>@d</td>
<td>PF13</td>
</tr>
<tr>
<td>@e</td>
<td>PF14</td>
</tr>
</tbody>
</table>
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

\[
[\text{boolean}] = \text{Display.Wait}([\text{timeout}])
\]

Remarks

The \text{timeout} parameter is optional and is measured in milliseconds. This method returns 'True' if the wait is successful and 'False' if the \text{timeout} period expires.

This method uses alternatively the \text{WaitForConnect}, \text{WaitForUnlock} and \text{WaitForNewScreen} methods, according to the current state of the host (disconnected, locked or unlocked).

If you specify a value in the \text{timeout} parameter, the value of the WaitTimeout property will not have any effect.

Related Topics

- \text{WaitForConnect Method}
- \text{WaitForUnlock Method}
- \text{WaitForNewScreen Method}

5.1.1.2.8 WaitFor

The WaitFor method waits for a screen that contains the specified string.

Syntax

\[
[\text{boolean}] = \text{Display.WaitFor(string, [timeout])}
\]
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

```csharp
[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

```csharp
[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
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] = \text{Display}.\text{WaitForScreen}([\text{timeout}])

Remarks

This method is similar to the \text{WaitForNewScreen} method, except that the screen to wait for can be any screen and not just a new one.

The \text{timeout} parameter is optional and is measured in milliseconds. This method returns 'True' if the wait is successful and 'False' if the \text{timeout} period expires.

If you specify a value in the \text{timeout} parameter, the value of the \text{WaitTimeout} property will not have any effect.

Related Topics

- \text{WaitForNewScreen}

5.1.1.2.12 WaitForNewScreen

The WaitForNewScreen method waits until a new screen is displayed.

Syntax

\[boolean] = \text{Display}.\text{WaitForNewScreen}([\text{timeout}])

Remarks

This method is similar to the \text{WaitForScreen} method, except that in WaitForNewScreen, the screen to wait for has to be a new one.

The \text{timeout} parameter is optional and is measured in milliseconds. This method returns 'True' if the wait is successful and 'False' if the \text{timeout} period expires.

If you specify a value in the \text{timeout} parameter, the value of the \text{WaitTimeout} property will not have any effect.

Related Topics

- \text{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)](#)
5.2.1.1 Cancel File Transfer (92)

The Cancel File Transfer function causes any current EHLLAPI initiated Send File or Receive File for the specified session to immediately return.

<table>
<thead>
<tr>
<th>3270</th>
<th>5250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Prerequisite Calls**

Send File (90) or Receive File (91)

**Call Parameters**

<table>
<thead>
<tr>
<th>Function Number</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Must be 92</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data String</th>
<th>1-character short name of the host presentation space. A blank or null indicates request for updates to the host-connected presentation space</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>4 is implied</td>
</tr>
<tr>
<td>PS Position</td>
<td>NA</td>
</tr>
</tbody>
</table>

The calling data structure contains these elements

<table>
<thead>
<tr>
<th>Byte</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A 1-character presentation space short name (PSID)</td>
</tr>
<tr>
<td>2-4</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

**Return Parameters**

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The function was successful</td>
</tr>
</tbody>
</table>
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

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.

<table>
<thead>
<tr>
<th>3270</th>
<th>5250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Prerequisite Calls

Connect Window Services (101)

Call Parameters

<table>
<thead>
<tr>
<th>Function Number</th>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Must be 106</td>
<td></td>
</tr>
<tr>
<td>Data String</td>
<td>See the following table</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>Must be specified (See note.)</td>
<td>Must be 68</td>
</tr>
<tr>
<td>PS Position</td>
<td>NA</td>
<td></td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Byte</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A 1-character presentation space short name (PSID)</td>
</tr>
<tr>
<td>2-4</td>
<td>Reserved</td>
</tr>
<tr>
<td>2</td>
<td>A change request option value, select one of:</td>
</tr>
<tr>
<td>3-63</td>
<td>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.</td>
</tr>
</tbody>
</table>
Programming Reference 283

character followed by a NULL character.

Reserved

67-68

Return Parameters

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The Change PS Window Name function was successful.</td>
</tr>
<tr>
<td>1</td>
<td>An incorrect host presentation space short session ID was specified, or the host presentation space was not connected.</td>
</tr>
<tr>
<td>2</td>
<td>An error was made in specifying parameters.</td>
</tr>
<tr>
<td>9</td>
<td>A system error occurred.</td>
</tr>
<tr>
<td>12</td>
<td>The session stopped.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>3270</th>
<th>5250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Prerequisite Calls

Connect Window Services (101)

Call Parameters

<table>
<thead>
<tr>
<th></th>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Number</td>
<td>Must be 105</td>
<td></td>
</tr>
<tr>
<td>Data String</td>
<td>See the following table</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>Normally 4-63</td>
<td>Must be 68</td>
</tr>
<tr>
<td>PS Position</td>
<td>NA</td>
<td></td>
</tr>
</tbody>
</table>
The calling data string can contain:

<table>
<thead>
<tr>
<th>Byte</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A 1-character presentation space short name (PSID)</td>
</tr>
<tr>
<td>2</td>
<td>A change request option; select:</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>3-63</td>
<td>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.</td>
</tr>
<tr>
<td>67-68</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

Return Parameters

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The Change Switch List LT Name function was successful.</td>
</tr>
<tr>
<td>1</td>
<td>An incorrect host presentation space short session ID was specified, or the host presentation space was not connected.</td>
</tr>
<tr>
<td>2</td>
<td>An error was made in specifying parameters.</td>
</tr>
<tr>
<td>9</td>
<td>A system error occurred.</td>
</tr>
<tr>
<td>12</td>
<td>The session stopped.</td>
</tr>
</tbody>
</table>

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 EHL/LAPI application program and the host presentation space.

<table>
<thead>
<tr>
<th>3270</th>
<th>5250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Prerequisite Calls

There are no prerequisite calls for this function.

Call Parameters

<table>
<thead>
<tr>
<th></th>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Number</td>
<td>Must be 1</td>
<td></td>
</tr>
<tr>
<td>Data String</td>
<td>1-character short name of the host presentation space</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>1 is implied</td>
<td>Must be 4</td>
</tr>
<tr>
<td>PS Position</td>
<td>NA</td>
<td></td>
</tr>
</tbody>
</table>

The calling data string can contain:

<table>
<thead>
<tr>
<th>Byte</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1-character presentation space short name (PSID)</td>
</tr>
<tr>
<td>2-4</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The Connect Presentation Space function was successful; the host presentation space is unlocked and ready for input.</td>
</tr>
<tr>
<td>1</td>
<td>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.</td>
</tr>
<tr>
<td>4</td>
<td>Successful connection was achieved, but the host presentation space is busy.</td>
</tr>
<tr>
<td>5</td>
<td>Successful connection was achieved, but the host presentation space is locked (input inhibited).</td>
</tr>
<tr>
<td>9</td>
<td>A system error was encountered.</td>
</tr>
<tr>
<td>11</td>
<td>This resource is unavailable. The host presentation space is already being used by another system function.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th></th>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>3270</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Prerequisite Calls**

There are no prerequisite calls for this function.

**Call Parameters**

<table>
<thead>
<tr>
<th></th>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Number</td>
<td>Must be 101</td>
<td></td>
</tr>
<tr>
<td>Data String</td>
<td>1-character short session ID of the host presentation space</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>1 is implied</td>
<td>Must be 4</td>
</tr>
<tr>
<td>PS Position</td>
<td>NA</td>
<td></td>
</tr>
</tbody>
</table>

The calling data string can contain:

<table>
<thead>
<tr>
<th>Byte</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A 1-character presentation space short name (PSID)</td>
</tr>
<tr>
<td>2-4</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

**Return Parameters**
<table>
<thead>
<tr>
<th>Return Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The Connect Window Services function was successful.</td>
</tr>
<tr>
<td>1</td>
<td>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.</td>
</tr>
<tr>
<td>9</td>
<td>A system error occurred.</td>
</tr>
<tr>
<td>10</td>
<td>The function is not supported by the emulation program.</td>
</tr>
<tr>
<td>11</td>
<td>This resource is unavailable. The host presentation space is already being used by another system function.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>3270</th>
<th>5250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Prerequisite Calls

There are no prerequisite calls for this function.

Call Parameters

<table>
<thead>
<tr>
<th>Function Number</th>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data String</td>
<td>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).</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>PS Position</td>
<td>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</td>
<td></td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>Byte</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>A 1-character presentation space short name (PSID)</td>
</tr>
<tr>
<td>2-4</td>
<td>Reserved</td>
</tr>
<tr>
<td>2</td>
<td>Convert option P or R</td>
</tr>
<tr>
<td>6-8</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

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:
<table>
<thead>
<tr>
<th>Status Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>This is an incorrect PS position or column.</td>
</tr>
<tr>
<td>&gt;0</td>
<td>This is the PS position or column.</td>
</tr>
<tr>
<td>9998</td>
<td>An incorrect host presentation space ID was specified or a system error occurred.</td>
</tr>
<tr>
<td>9999</td>
<td>Character 2 in the data string is not P or R.</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>3270</th>
<th>5250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Prerequisite Calls**

**Connect Presentation Space (1)**

**Call Parameters**

<table>
<thead>
<tr>
<th></th>
<th>Standard Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Number</td>
<td>Must be 34</td>
</tr>
<tr>
<td>Data String</td>
<td>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.</td>
</tr>
<tr>
<td>Length</td>
<td>Number of bytes to copy (the length of the data string).</td>
</tr>
<tr>
<td>PS Position</td>
<td>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.</td>
</tr>
</tbody>
</table>

**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:

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The Copy Field to String function was successful.</td>
</tr>
<tr>
<td>1</td>
<td>Your program is not connected to a host session.</td>
</tr>
<tr>
<td>2</td>
<td>An error was made in specifying parameters.</td>
</tr>
<tr>
<td>6</td>
<td>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.</td>
</tr>
<tr>
<td>7</td>
<td>The host presentation space position is not valid.</td>
</tr>
<tr>
<td>9</td>
<td>A system error was encountered.</td>
</tr>
<tr>
<td>24</td>
<td>Unformatted host presentation space.</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th></th>
<th>3270</th>
<th>5250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>

Prerequisite Calls

Connect Presentation Space (1)

Call Parameters

<table>
<thead>
<tr>
<th></th>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Number</td>
<td>Must be 13</td>
<td></td>
</tr>
<tr>
<td>Data String</td>
<td>Preallocated target data string</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>103</td>
<td>104</td>
</tr>
<tr>
<td>PS Position</td>
<td>NA</td>
<td></td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OIA data is returned. The target presentation space is unlocked.</td>
</tr>
<tr>
<td>1</td>
<td>Your program is not connected to a host session.</td>
</tr>
<tr>
<td>2</td>
<td>An error was made in specifying string length. OIA data was not returned.</td>
</tr>
<tr>
<td>4</td>
<td>OIA data is returned. The target presentation space is busy.</td>
</tr>
<tr>
<td>5</td>
<td>OIA data is returned. The target presentation space is locked. (Input inhibited)</td>
</tr>
<tr>
<td>9</td>
<td>An internal system error was encountered. OIA data was not returned.</td>
</tr>
</tbody>
</table>

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.
2. 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 EHLLAPI 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.

<table>
<thead>
<tr>
<th>3270</th>
<th>5250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Prerequisite Calls

Connect Presentation Space (1)

Call Parameters

<table>
<thead>
<tr>
<th>Function Number</th>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data String</td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>NA (the length of the host presentation space is implied).</td>
<td></td>
</tr>
<tr>
<td>PS Position</td>
<td>NA.</td>
<td></td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The host presentation space contents were copied to the application program. The target presentation space was active, and the keyboard was unlocked.</td>
</tr>
<tr>
<td>1</td>
<td>Your program is not connected to a host session.</td>
</tr>
<tr>
<td>4</td>
<td>The host presentation space contents were copied. The connected host presentation space was waiting for host response.</td>
</tr>
<tr>
<td>5</td>
<td>The host presentation space was copied. The keyboard was locked.</td>
</tr>
<tr>
<td>9</td>
<td>A system error was encountered.</td>
</tr>
</tbody>
</table>

**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.
3. 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
4. If you want to copy only a portion of the host presentation space, use the Copy Presentation Space to String (8) function.
5. 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 EHLLAPI 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.

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3270</td>
<td>Yes</td>
</tr>
<tr>
<td>5250</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Prerequisite Calls**

Connect Presentation Space (1)

**Call Parameters**

<table>
<thead>
<tr>
<th>Function Parameter</th>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Number</td>
<td>Must be 8</td>
<td></td>
</tr>
<tr>
<td>Data String</td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>Length of the target data string.</td>
<td></td>
</tr>
<tr>
<td>PS Position</td>
<td>Position within the host presentation space of the first byte in your target data string.</td>
<td></td>
</tr>
</tbody>
</table>

**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:

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The host presentation space contents were copied to the application program. The target presentation space was active, and the keyboard was unlocked.</td>
</tr>
<tr>
<td>1</td>
<td>Your program is not connected to a host session.</td>
</tr>
<tr>
<td>2</td>
<td>An error was made in specifying string length, or the sum of (Length - 1) + PS position is greater</td>
</tr>
</tbody>
</table>
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
   - 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.

<table>
<thead>
<tr>
<th>3270</th>
<th>5250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Prerequisite Calls

Connect Presentation Space (1)
Call Parameters

<table>
<thead>
<tr>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Number</td>
<td>Must be 33</td>
</tr>
<tr>
<td>Data String</td>
<td>String containing the data to be transferred to a target field in the host presentation space.</td>
</tr>
<tr>
<td>Length</td>
<td>Length, in number of bytes, of the source data string. Overridden if in EOT mode.</td>
</tr>
<tr>
<td>PS Position</td>
<td>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.</td>
</tr>
</tbody>
</table>

Return Parameters

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The Copy String to Field function was successful.</td>
</tr>
<tr>
<td>1</td>
<td>Your program is not connected to a host session.</td>
</tr>
<tr>
<td>2</td>
<td>Parameter error or zero length for copy.</td>
</tr>
<tr>
<td>5</td>
<td>The target field was protected or inhibited, or incorrect data was sent to the target field (such as a field attribute).</td>
</tr>
<tr>
<td>6</td>
<td>Copy was completed, but data is truncated.</td>
</tr>
<tr>
<td>7</td>
<td>The host presentation space position is not valid.</td>
</tr>
<tr>
<td>9</td>
<td>A system error was encountered.</td>
</tr>
<tr>
<td>24</td>
<td>Unformatted host presentation space.</td>
</tr>
</tbody>
</table>

Remarks

1. The Copy String to Field function is affected by the following options:
   - STRLEN/STREOT
   - EOT
   - EAB/NOEAB
   - XLATE/NOXLATE
   - PUTEAB/NOPUTEAB
2. 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.
3. The keyboard mnemonics (see Send Key (3) function) cannot be sent using the Copy String to Field function.
4. The first byte of the data to be transferred is always placed at the beginning of the field that contains the specified PS position.
5. 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.
Prerequisite Calls

Connect Presentation Space (1)

Call Parameters

<table>
<thead>
<tr>
<th></th>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Number</td>
<td>Must be 15.</td>
<td></td>
</tr>
<tr>
<td>Data String</td>
<td>String of ASCII data to be copied into the host presentation space.</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>Length, in number of bytes, of the source data string. Overridden if in EOT mode.</td>
<td></td>
</tr>
<tr>
<td>PS Position</td>
<td>Position in the host presentation space to begin the copy, a value between 1 and the configured size of your host presentation space.</td>
<td></td>
</tr>
</tbody>
</table>

Return Parameters

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The Copy String to Presentation Space function was successful.</td>
</tr>
<tr>
<td>1</td>
<td>Your program is not connected to a host session.</td>
</tr>
<tr>
<td>2</td>
<td>Parameter error or zero length for copy.</td>
</tr>
<tr>
<td>5</td>
<td>The target presentation space is protected or inhibited, or incorrect data was sent to the target presentation space (such as a field attribute byte).</td>
</tr>
<tr>
<td>6</td>
<td>The copy was completed, but the data was truncated.</td>
</tr>
<tr>
<td>7</td>
<td>The host presentation space position is not valid.</td>
</tr>
<tr>
<td>9</td>
<td>A system error was encountered.</td>
</tr>
</tbody>
</table>

Remarks

1. The Copy String to Presentation Space function is affected by the following options:
   - STRLEN/STREOT
   - EOT
   - EAB/NOEAB
   - XLAITE/NOXLAITE
   - PUTEAB/NOPUTEAB
   - EAD/NOEAD (for DBCS only)
   - NOSO/SPACESO/SO (for DBCS only)
   - EXTEND_PS/NOEXTEND_PS
2. The keyboard mnemonics (see Send Key (3) function) cannot be sent using the Copy String to Presentation Space function.
3. 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.
4. 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.

<table>
<thead>
<tr>
<th>3270</th>
<th>5250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Prerequisite Calls

Connect Presentation Space (1)

Call Parameters

<table>
<thead>
<tr>
<th></th>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Number</td>
<td>Must be 2</td>
<td></td>
</tr>
<tr>
<td>Data String</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>PS Position</td>
<td>NA</td>
<td></td>
</tr>
</tbody>
</table>

Return Parameters

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The Disconnect Presentation Space function was successful.</td>
</tr>
<tr>
<td>1</td>
<td>Your program was not currently connected to the host presentation space.</td>
</tr>
<tr>
<td>9</td>
<td>A system error was encountered.</td>
</tr>
</tbody>
</table>

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 EHLLAPI program and the specified host presentation space window.

<table>
<thead>
<tr>
<th></th>
<th>3270</th>
<th>5250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>

Prerequisite Calls

Connect Window Services (101)

Call Parameters

<table>
<thead>
<tr>
<th></th>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Number</td>
<td>Must be 102</td>
<td></td>
</tr>
<tr>
<td>Data String</td>
<td>See the following table</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>PS Position</td>
<td>NA</td>
<td></td>
</tr>
</tbody>
</table>

Data String Contents

<table>
<thead>
<tr>
<th>Byte</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Standard  Enhanced</td>
</tr>
<tr>
<td>1</td>
<td>1 A 1-character presentation space short name (PSID)</td>
</tr>
<tr>
<td>2-4</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

Return Parameters

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The Disconnect Window Service function was successful.</td>
</tr>
<tr>
<td>1</td>
<td>Your program is not connected for Window Services.</td>
</tr>
<tr>
<td>9</td>
<td>A system error occurred.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>3270</th>
<th>5250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Prerequisite Calls

Connect Presentation Space (1)

Call Parameters

<table>
<thead>
<tr>
<th></th>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Number</td>
<td>Must be 32</td>
<td></td>
</tr>
<tr>
<td>Data String</td>
<td>See the following table</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>PS Position</td>
<td>See note</td>
<td></td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>[RB][RB] or T [RB]</td>
<td>This field</td>
</tr>
<tr>
<td>P [RB]</td>
<td>The previous field, either protected or unprotected.</td>
</tr>
<tr>
<td>N [RB]</td>
<td>The next field, either protected or unprotected</td>
</tr>
<tr>
<td>NP</td>
<td>The next protected field</td>
</tr>
<tr>
<td>NU</td>
<td>The next unprotected field</td>
</tr>
<tr>
<td>PP</td>
<td>The previous protected field</td>
</tr>
<tr>
<td>PU</td>
<td>The previous unprotected field</td>
</tr>
</tbody>
</table>

Note:

[RB] represents a required blank.

Return Parameters

This function returns a length and a return code.

Length
The following lengths are valid:

<table>
<thead>
<tr>
<th>Length</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>= 0</td>
<td>When return code = 28, field length is 0. When return code = 24, host presentation space is not field formatted.</td>
</tr>
<tr>
<td>&gt; 0</td>
<td>Required field length in the host presentation space.</td>
</tr>
</tbody>
</table>

**Return Code**

The following codes are defined:

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The Find Field Length function was successful.</td>
</tr>
<tr>
<td>1</td>
<td>Your program is not connected to a host session.</td>
</tr>
<tr>
<td>2</td>
<td>A parameter error was encountered.</td>
</tr>
<tr>
<td>7</td>
<td>The host presentation space position is not valid.</td>
</tr>
<tr>
<td>9</td>
<td>A system error was encountered.</td>
</tr>
<tr>
<td>24</td>
<td>No such field was found.</td>
</tr>
<tr>
<td>28</td>
<td>Field length of 0 bytes.</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>3270</th>
<th>5250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Prerequisite Calls**

Connect Presentation Space (1)

**Call Parameters**

<table>
<thead>
<tr>
<th></th>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Number</td>
<td>Must be 31</td>
<td></td>
</tr>
<tr>
<td>Data String</td>
<td>See the following table</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>PS Position</td>
<td>See note</td>
<td></td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>[RB][RB] or T [RB]</td>
<td>This field</td>
</tr>
<tr>
<td>P [RB]</td>
<td>The previous field, either protected or unprotected</td>
</tr>
<tr>
<td>N [RB]</td>
<td>The next field, either protected or unprotected</td>
</tr>
<tr>
<td>NP</td>
<td>The next protected field</td>
</tr>
<tr>
<td>NU</td>
<td>The next unprotected field</td>
</tr>
<tr>
<td>PP</td>
<td>The previous protected field</td>
</tr>
<tr>
<td>PU</td>
<td>The previous unprotected field</td>
</tr>
</tbody>
</table>

Note: [RB] represents a required blank.

**Return Parameters**

This function returns a length and a return code.

**Length**

The following lengths are valid:

<table>
<thead>
<tr>
<th>Length</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>= 0</td>
<td>When return code = 28, field length is 0. When return code = 24, host presentation space is not field-formatted.</td>
</tr>
<tr>
<td>&gt; 0</td>
<td>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.</td>
</tr>
</tbody>
</table>

**Return Code**

The following codes are defined:

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The Find Field Position function was successful.</td>
</tr>
<tr>
<td>1</td>
<td>Your program is not connected to a host session.</td>
</tr>
<tr>
<td>2</td>
<td>A parameter error was encountered.</td>
</tr>
<tr>
<td>7</td>
<td>The host presentation space position is not valid.</td>
</tr>
<tr>
<td>9</td>
<td>A system error was encountered.</td>
</tr>
<tr>
<td>24</td>
<td>No such field was found.</td>
</tr>
<tr>
<td>28</td>
<td>Field length of 0 bytes.</td>
</tr>
</tbody>
</table>

**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 EHLLAPI 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.

<table>
<thead>
<tr>
<th>3270</th>
<th>5250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Prerequisite Calls**

Start Keystroke Intercept (50)

**Call Parameters**

<table>
<thead>
<tr>
<th>Function Number</th>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data String</td>
<td>See the following table</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>8</td>
<td>12</td>
</tr>
<tr>
<td>PS Position</td>
<td>NA</td>
<td></td>
</tr>
</tbody>
</table>

**Data String Contents**

<table>
<thead>
<tr>
<th>Byte</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>One of the following values:</td>
</tr>
<tr>
<td></td>
<td>- A 1-character presentation space short name (PSID)</td>
</tr>
<tr>
<td></td>
<td>- A blank or null indicating a function call for the host-connected presentation</td>
</tr>
<tr>
<td>2-4</td>
<td>Reserved</td>
</tr>
<tr>
<td>2-8</td>
<td>Blanks that hold space for the symbolic representation of the requested data</td>
</tr>
<tr>
<td>5-11</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

**Return Parameters**

This function returns a data string and a return code.

**Data String**

See the following table:

<table>
<thead>
<tr>
<th>Byte</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>One of the following values:</td>
</tr>
<tr>
<td></td>
<td>- A 1-character presentation space short name (PSID)</td>
</tr>
<tr>
<td></td>
<td>- A blank or null indicating a function call for the host-connected presentation</td>
</tr>
<tr>
<td>2-4</td>
<td>Reserved</td>
</tr>
<tr>
<td>Return Code</td>
<td>Explanation</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>0</td>
<td>The Get Key function was successful.</td>
</tr>
<tr>
<td>1</td>
<td>An incorrect presentation space was specified.</td>
</tr>
<tr>
<td>5</td>
<td>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.</td>
</tr>
<tr>
<td>8</td>
<td>No prior Start Keystroke Intercept (50) function was called for this presentation space.</td>
</tr>
<tr>
<td>9</td>
<td>A system error was encountered.</td>
</tr>
<tr>
<td>20</td>
<td>An undefined key combination was typed.</td>
</tr>
<tr>
<td>25</td>
<td>The requested keystrokes are not available on the input queue.</td>
</tr>
<tr>
<td>31</td>
<td>Keystroke queue overflowed and keystrokes were lost.</td>
</tr>
</tbody>
</table>

### Remarks

1. 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.

2. 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
3. Keystrokes arrive asynchronously and are enqueued in the keystroke queue that you have provided in your EHLLAPI 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 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.

<table>
<thead>
<tr>
<th>3270</th>
<th>5250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

**Prerequisite Calls**

Connect to Presentation Space (1)

**Call Parameters**

<table>
<thead>
<tr>
<th></th>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Number</td>
<td>Must be 60</td>
<td></td>
</tr>
<tr>
<td>Data String</td>
<td>See the following table</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>Must be 3</td>
<td>Must be 8</td>
</tr>
<tr>
<td>PS Position</td>
<td>NA</td>
<td></td>
</tr>
</tbody>
</table>

**Data String Contents**

<table>
<thead>
<tr>
<th>Byte</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Enhanced</td>
<td></td>
</tr>
<tr>
<td>1 1</td>
<td>A 1-character presentation space short name (PSID).</td>
</tr>
<tr>
<td>2-4</td>
<td>Reserved.</td>
</tr>
</tbody>
</table>
| 2 5 | One of the following characters:  
|    | • L to lock the API.  
|    | • U to unlock the API. |
One of the following characters:
• 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.

<table>
<thead>
<tr>
<th>Return Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Return Code</strong></td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>43</td>
</tr>
</tbody>
</table>

Remarks

The following EHLLAPI 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. EHLLAPI 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
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.

<table>
<thead>
<tr>
<th></th>
<th>3270</th>
<th>5250</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

Prerequisite Calls

Connect Window Services (101)

Call Parameters

<table>
<thead>
<tr>
<th></th>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Number</td>
<td>Must be 61</td>
<td></td>
</tr>
<tr>
<td>Data String</td>
<td>See the following table.</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>Must be 3</td>
<td>Must be 8</td>
</tr>
<tr>
<td>PS Position</td>
<td>NA</td>
<td></td>
</tr>
</tbody>
</table>

Data String Contents

<table>
<thead>
<tr>
<th>Byte</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Standard Enhanced</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2-4</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>L to lock the API.</td>
</tr>
<tr>
<td></td>
<td>U to unlock the API.</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>R to return if the presentation space is already locked by an application.</td>
</tr>
<tr>
<td></td>
<td>Q to queue the Lock request if the presentation space is already locked by an application.</td>
</tr>
<tr>
<td></td>
<td>7-8</td>
</tr>
</tbody>
</table>
Return Parameters

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The Lock Window Services API function was successful.</td>
</tr>
<tr>
<td>1</td>
<td>An incorrect host presentation space short session ID was specified or was not connected.</td>
</tr>
<tr>
<td>2</td>
<td>An error was made in specifying parameters.</td>
</tr>
<tr>
<td>9</td>
<td>A system error was encountered.</td>
</tr>
<tr>
<td>38</td>
<td>Requested function was not complete.</td>
</tr>
<tr>
<td>43</td>
<td>The API was already locked by another EHLLAPI application (on LOCK) or API not locked (on UNLOCK).</td>
</tr>
</tbody>
</table>

Remarks

The following EHLLAPI 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.

<table>
<thead>
<tr>
<th>3270</th>
<th>5250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Prerequisite Calls

There are no prerequisite calls for this function.

Call Parameters
### Function Number

Must be 18

### Data String

NA

### Length

Contains the pause duration in half-second increments

### PS Position

NA

## Return Parameters

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The wait duration has expired.</td>
</tr>
<tr>
<td>9</td>
<td>An internal system error was encountered. The time results are unpredictable.</td>
</tr>
<tr>
<td>26</td>
<td>The host session presentation space or OIA has been updated. Use the Query Host Update (24) function to get more information.</td>
</tr>
</tbody>
</table>

## 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.

<table>
<thead>
<tr>
<th>3270</th>
<th>5250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Prerequisite Calls**

Start Keystroke Intercept (50)

**Call Parameters**

<table>
<thead>
<tr>
<th></th>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Number</td>
<td>Must be 52</td>
<td></td>
</tr>
<tr>
<td>Data String</td>
<td>See the following table</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>Must be 2</td>
<td>Must be 8</td>
</tr>
<tr>
<td>PS Position</td>
<td>NA</td>
<td></td>
</tr>
</tbody>
</table>

The calling data string can contain:

<table>
<thead>
<tr>
<th>Byte</th>
<th>Definition</th>
<th>Standard</th>
<th>Enhanced</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>One of the following values:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• The 1-letter short name of the presentation space.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• A blank or null indicating a function call for the host-connected presentation space.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-4</td>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>One of the following characters:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• A for accepted keystroke.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• R for rejected keystroke.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6-8</td>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Remarks**

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The Post Intercept Status function was successful.</td>
</tr>
<tr>
<td>1</td>
<td>An incorrect presentation space was specified.</td>
</tr>
<tr>
<td>2</td>
<td>An incorrect session option was specified.</td>
</tr>
<tr>
<td>8</td>
<td>No prior Start Keystroke Intercept (50) function was called for this presentation space ID.</td>
</tr>
<tr>
<td>9</td>
<td>A system error was encountered.</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>3270</th>
<th>5250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Prerequisite Calls**

*Connect Presentation Space (1)*

**Call Parameters**

<table>
<thead>
<tr>
<th></th>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Number</td>
<td>Must be 45.</td>
<td></td>
</tr>
<tr>
<td>Data String</td>
<td>8 bytes long character string.</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>8 is implied.</td>
<td></td>
</tr>
<tr>
<td>PS Position</td>
<td>Identifies the target. This can be the PS position of any byte within the target field.</td>
<td></td>
</tr>
</tbody>
</table>

The calling data string can contain:

<table>
<thead>
<tr>
<th>Byte</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-8</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

**Return Parameters**

This function returns a data string and a return code.

**Data String**

The function returns the following data string.

<table>
<thead>
<tr>
<th>Byte</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-6</td>
<td>Reserved</td>
</tr>
<tr>
<td>7-8</td>
<td>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.</td>
</tr>
</tbody>
</table>

**Return Code**

The following return codes are defined:

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The Query Additional Field Attribute was successful.</td>
</tr>
<tr>
<td>1</td>
<td>Your program is not currently connected to a host session.</td>
</tr>
<tr>
<td>7</td>
<td>The host presentation space position is not valid.</td>
</tr>
</tbody>
</table>
5.2.1.23 Query Close Intercept (42)

The Query Close Intercept function allows the application to determine if the close option was selected.

<table>
<thead>
<tr>
<th>3270</th>
<th>5250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Prerequisite Calls**

Start Close Intercept (41)

**Call Parameters**

<table>
<thead>
<tr>
<th>Function Number</th>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Must be 42</td>
<td></td>
</tr>
<tr>
<td>Data String</td>
<td>See the following table.</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>Must be 1</td>
<td>Must be 4</td>
</tr>
<tr>
<td>PS Position</td>
<td>NA</td>
<td></td>
</tr>
</tbody>
</table>

The calling data string can contain:

<table>
<thead>
<tr>
<th>Byte</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1-character short session ID of the host presentation space, or a blank or null indicating request for querying the host-connected session</td>
</tr>
<tr>
<td>2-4</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

**Return Parameters**

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>A close intercept event did not occur.</td>
</tr>
<tr>
<td>1</td>
<td>The presentation source was not valid.</td>
</tr>
<tr>
<td>2</td>
<td>An error was made in specifying parameters.</td>
</tr>
<tr>
<td>8</td>
<td>No prior Start Close Intercept (41) function was called for this host presentation space.</td>
</tr>
<tr>
<td>9</td>
<td>A system error occurred.</td>
</tr>
<tr>
<td>12</td>
<td>The session stopped.</td>
</tr>
<tr>
<td>26</td>
<td>A close intercept occurred since the last query close intercept call.</td>
</tr>
</tbody>
</table>

5.2.1.24 Query Communication Event (81)

The Query Communication Event function lets the EHLLAPI program determine whether any communication events have occurred.
Prerequisite Calls

Start Communication Notification (80)

Call Parameters

<table>
<thead>
<tr>
<th>Enhanced Interface</th>
<th>Function Number</th>
<th>Data String</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Must be 81</td>
<td>1-character short name of the host presentation space or a blank or null indicating request for updates to the host-connected presentation space</td>
<td>4 is implied</td>
</tr>
</tbody>
</table>

The calling data structure contains these elements:

<table>
<thead>
<tr>
<th>Byte</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A 1-character presentation space short name (PSID)</td>
</tr>
<tr>
<td>2-4</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

Return Parameters

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The function was successful</td>
</tr>
<tr>
<td>1</td>
<td>An incorrect PSID was specified</td>
</tr>
<tr>
<td>8</td>
<td>No prior call to Start Communication Notification (80) function was called for the PSID</td>
</tr>
<tr>
<td>9</td>
<td>A system error was encountered</td>
</tr>
<tr>
<td>21</td>
<td>The indicated PSID was connected</td>
</tr>
<tr>
<td>22</td>
<td>The Indicated PSID was disconnected</td>
</tr>
</tbody>
</table>

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.

Prerequisite Calls

Connect Presentation Space (1)

Call Parameters
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:

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The Query Cursor Location function was successful.</td>
</tr>
<tr>
<td>1</td>
<td>Your program is not currently connected to a host session.</td>
</tr>
<tr>
<td>9</td>
<td>A system error was encountered.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>3270</th>
<th>5250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Prerequisite Calls

Connect Presentation Space (1)

Call Parameters

<table>
<thead>
<tr>
<th></th>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Number</td>
<td>Must be 7</td>
<td></td>
</tr>
<tr>
<td>Data String</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>PS Position</td>
<td>NA</td>
<td></td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The Query Field Attribute was successful.</td>
</tr>
<tr>
<td>1</td>
<td>Your program is not currently connected to a host session.</td>
</tr>
<tr>
<td>7</td>
<td>The host presentation space position is not valid.</td>
</tr>
<tr>
<td>9</td>
<td>A system error was encountered.</td>
</tr>
<tr>
<td>24</td>
<td>Attribute byte not found or unformatted host presentation space.</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Bit Position</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-1</td>
<td>Both = 1, field attribute byte</td>
</tr>
<tr>
<td>2</td>
<td>Unprotected/protected</td>
</tr>
<tr>
<td>3</td>
<td>A/N</td>
</tr>
<tr>
<td>4-5</td>
<td>I/SPD</td>
</tr>
<tr>
<td>6</td>
<td>Reserved</td>
</tr>
<tr>
<td>7</td>
<td>MDT</td>
</tr>
</tbody>
</table>

5250 field attributes:

<table>
<thead>
<tr>
<th>Bit Position</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Field attribute flag</td>
</tr>
<tr>
<td>1</td>
<td>Visibility</td>
</tr>
</tbody>
</table>
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).

<table>
<thead>
<tr>
<th>3270</th>
<th>5250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Prerequisite Calls

Start Host Notification (23)

Call Parameters

<table>
<thead>
<tr>
<th>Function Number</th>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data String</td>
<td>1-character short name of the host presentation space, or a blank or null indicating request for updates to host-connected presentation space</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>1 is implied</td>
<td>4 is implied</td>
</tr>
<tr>
<td>PS Position</td>
<td>NA</td>
<td></td>
</tr>
</tbody>
</table>

The calling data string can contain:

<table>
<thead>
<tr>
<th>Byte</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td>Enhanced</td>
</tr>
<tr>
<td>1</td>
<td>1 1-character presentation space short name (PSID)</td>
</tr>
</tbody>
</table>
Return Parameters

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No updates have been made since the last call.</td>
</tr>
<tr>
<td>1</td>
<td>An incorrect host presentation space was specified.</td>
</tr>
<tr>
<td>8</td>
<td>No prior Start Host Notification (23) function was called for the host presentation space ID.</td>
</tr>
<tr>
<td>9</td>
<td>A system error was encountered.</td>
</tr>
<tr>
<td>21</td>
<td>The OIA was updated.</td>
</tr>
<tr>
<td>22</td>
<td>The presentation space was updated.</td>
</tr>
<tr>
<td>23</td>
<td>Both the OIA and the host presentation space were updated.</td>
</tr>
<tr>
<td>44</td>
<td>Printing has completed in the printer session.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>3270</th>
<th>5250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Prerequisite Calls

There are no prerequisite calls for this function.

Call Parameters

<table>
<thead>
<tr>
<th></th>
<th>16-bit</th>
<th>32-bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Number</td>
<td>Must be 22.</td>
<td></td>
</tr>
</tbody>
</table>
| 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:  
  - 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

<table>
<thead>
<tr>
<th>Byte</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1-4</td>
</tr>
<tr>
<td>2-9</td>
<td>5-11</td>
</tr>
<tr>
<td>10</td>
<td>13</td>
</tr>
<tr>
<td>11</td>
<td>14</td>
</tr>
<tr>
<td>12-13</td>
<td>15-16</td>
</tr>
<tr>
<td>14-15</td>
<td>17-18</td>
</tr>
<tr>
<td>16-17</td>
<td>19-20</td>
</tr>
<tr>
<td>18</td>
<td></td>
</tr>
</tbody>
</table>

#### Return Code

The following codes are defined:

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The Query Session Status function was successful.</td>
</tr>
<tr>
<td>1</td>
<td>An incorrect host presentation space was specified.</td>
</tr>
<tr>
<td>2</td>
<td>An incorrect string length was made.</td>
</tr>
<tr>
<td>9</td>
<td>A system error was encountered.</td>
</tr>
</tbody>
</table>

#### 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.

<table>
<thead>
<tr>
<th>3270</th>
<th>5250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

#### Prerequisite Calls

There are no prerequisite calls for this function.
Call Parameters

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Standard Interface</td>
</tr>
<tr>
<td>Function Number</td>
<td>Must be 10</td>
</tr>
<tr>
<td>Data String</td>
<td>Preallocated string of 16n bytes long (12n for 16-bit) (n = number of sessions) or more</td>
</tr>
<tr>
<td>Length</td>
<td>12n bytes</td>
</tr>
<tr>
<td>PS Position</td>
<td>NA</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Byte</th>
<th>Standard</th>
<th>Enhanced</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2-4</td>
<td>Reserved</td>
<td>Reserved</td>
</tr>
<tr>
<td>2-9</td>
<td>5-12</td>
<td>Session long name (same as profile name; or, if profile not set, same as short name)</td>
</tr>
<tr>
<td>10</td>
<td>13</td>
<td>Connection type H=host</td>
</tr>
<tr>
<td>14</td>
<td>Reserved</td>
<td>Reserved</td>
</tr>
<tr>
<td>11-12</td>
<td>15-16</td>
<td>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.</td>
</tr>
</tbody>
</table>

Length

The number of host sessions started.

Return Code

The following codes are defined:

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The Query Sessions function was successful.</td>
</tr>
<tr>
<td>2</td>
<td>An incorrect string length was made.</td>
</tr>
<tr>
<td>9</td>
<td>A system error was encountered.</td>
</tr>
</tbody>
</table>
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 EHLLAPI 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.

<table>
<thead>
<tr>
<th>3270</th>
<th>5250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Prerequisite Calls

There are no prerequisite calls for this function.

Call Parameters

<table>
<thead>
<tr>
<th>Function Number</th>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data String</td>
<td>Preallocated string of 35 bytes</td>
<td>36 bytes</td>
</tr>
<tr>
<td>Length</td>
<td>Must be 35</td>
<td>Must be 36</td>
</tr>
<tr>
<td>PS Position</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Byte</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>EHLLAPI version number</td>
</tr>
<tr>
<td>2-3</td>
<td>EHLLAPI level number</td>
</tr>
<tr>
<td>4-9</td>
<td>Reserved</td>
</tr>
<tr>
<td>10-12</td>
<td>Reserved</td>
</tr>
<tr>
<td>13</td>
<td>Hardware base, U=Unable to determine</td>
</tr>
<tr>
<td>14</td>
<td>Program type</td>
</tr>
</tbody>
</table>
### Return Code

The following codes are defined:

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The Query System function was successful; data string has been returned.</td>
</tr>
<tr>
<td>1</td>
<td>EHLLAPI is not loaded. (PC/3270 only)</td>
</tr>
<tr>
<td>2</td>
<td>An incorrect string length was specified. (PC/3270 only)</td>
</tr>
<tr>
<td>9</td>
<td>A system error was encountered.</td>
</tr>
</tbody>
</table>

### 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.

<table>
<thead>
<tr>
<th>3270</th>
<th>5250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### Prerequisite Calls

**Connect Windows Services (101)**

### Call Parameters

<table>
<thead>
<tr>
<th>Function Number</th>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Return Code</td>
<td>Explanation</td>
<td></td>
</tr>
<tr>
<td>Data String</td>
<td>1-character short session ID of the host presentation space</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>17 is implied</td>
<td>20 is implied</td>
</tr>
<tr>
<td>PS Position</td>
<td>NA</td>
<td></td>
</tr>
</tbody>
</table>

The calling data string can contain:
### Byte Definition

<table>
<thead>
<tr>
<th>Standard</th>
<th>Enhanced</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>One of the following values:</td>
</tr>
<tr>
<td></td>
<td>A 1-character presentation space short name (PSID)</td>
</tr>
<tr>
<td></td>
<td>A blank or null indicating a function call for the current connection presentation space</td>
</tr>
<tr>
<td>2-4</td>
<td>Reserved</td>
</tr>
<tr>
<td>2-17</td>
<td>5-20</td>
</tr>
<tr>
<td></td>
<td>Reserved</td>
</tr>
</tbody>
</table>

### Return Parameters

This function returns a data string and a return code.

<table>
<thead>
<tr>
<th>Standard</th>
<th>Enhanced</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>One of the following values:</td>
</tr>
<tr>
<td></td>
<td>A 1-character presentation space short session ID</td>
</tr>
<tr>
<td></td>
<td>A blank or null indicating a function call for the current connection presentation space</td>
</tr>
<tr>
<td>2-4</td>
<td>Reserved</td>
</tr>
<tr>
<td>2-17</td>
<td>5-20</td>
</tr>
<tr>
<td></td>
<td>Four 32-bit unsigned integers that return:</td>
</tr>
<tr>
<td>2-5</td>
<td>5-8</td>
</tr>
<tr>
<td></td>
<td>XLeft Long integer in pels of the left X coordinate of the rectangular window relative to the desktop window</td>
</tr>
<tr>
<td>6-9</td>
<td>9-12</td>
</tr>
<tr>
<td></td>
<td>YBottom Long integer in pels of the bottom Y coordinate of the rectangular window relative to the desktop window</td>
</tr>
<tr>
<td>10-13</td>
<td>13-15</td>
</tr>
<tr>
<td></td>
<td>XRight Long integer in pels of the right X coordinate of the rectangular window relative to the desktop window</td>
</tr>
<tr>
<td>14-17</td>
<td>16-20</td>
</tr>
<tr>
<td></td>
<td>YTop Long integer in pels of the top Y coordinate of the rectangular window relative to the desktop window</td>
</tr>
</tbody>
</table>

### Return Code

The following codes are defined:

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The Query Window Coordinates function was successful.</td>
</tr>
<tr>
<td>1</td>
<td>Your program was not currently connected to the host session.</td>
</tr>
<tr>
<td>9</td>
<td>A system error occurred.</td>
</tr>
<tr>
<td>12</td>
<td>The session stopped.</td>
</tr>
</tbody>
</table>

### 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.

<table>
<thead>
<tr>
<th>3270</th>
<th>5250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### Prerequisite Calls

There are no prerequisite calls for this function.
Call Parameters

<table>
<thead>
<tr>
<th></th>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Number</td>
<td>Must be 91.</td>
<td></td>
</tr>
<tr>
<td>Data String</td>
<td>Refer to the examples.</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>Length, in number of bytes, of the data string. Overridden if in EOT mode.</td>
<td></td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Host System</th>
<th>Common Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM/CMS</td>
<td>ASCII, JISCII, CRLF, APPEND, TIME n, CLEAR, NOCLEAR,</td>
</tr>
<tr>
<td></td>
<td>PROGRESS, QUIET</td>
</tr>
<tr>
<td>MVS/TSO</td>
<td>ASCII, JISCII, CRLF, APPEND, TIME (n), CLEAR, NOCLEAR,</td>
</tr>
<tr>
<td></td>
<td>QUIET, AVBLOCK</td>
</tr>
<tr>
<td>CICS</td>
<td>ASCII, JISCII, CRLF, NOCRLF, BINARY, TIME n, CLEAR,</td>
</tr>
<tr>
<td></td>
<td>NOCLEAR, PROGRESS, QUIET</td>
</tr>
<tr>
<td>i5/OS(TM) or OS/400(R)</td>
<td>ASCII, JISCII, CRLF, APPEND, TIME n, CLEAR, NOCLEAR,</td>
</tr>
<tr>
<td></td>
<td>PROGRESS, QUIET</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>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.</td>
</tr>
<tr>
<td>3</td>
<td>File transfer complete.</td>
</tr>
<tr>
<td>4</td>
<td>File transfer complete with segmented records.</td>
</tr>
<tr>
<td>9</td>
<td>A system error was encountered.</td>
</tr>
<tr>
<td>27</td>
<td>File transfer terminated because of either a Cancel button or the timeout set by the Set Session Parameter (9) function.</td>
</tr>
<tr>
<td>101</td>
<td>File transfer was successful (transfer to/from CICS).</td>
</tr>
</tbody>
</table>

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

1. 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.

Prerequisite Calls

Connect Presentation Space (1)

Call Parameters

<table>
<thead>
<tr>
<th></th>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Number</td>
<td>Must be 12</td>
<td></td>
</tr>
<tr>
<td>Data String</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>PS Position</td>
<td>NA</td>
<td></td>
</tr>
</tbody>
</table>

Return Parameters

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The Release function was successful.</td>
</tr>
<tr>
<td>1</td>
<td>Your program is not connected to a host session.</td>
</tr>
<tr>
<td>9</td>
<td>A system error was encountered.</td>
</tr>
</tbody>
</table>

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 EHLLAPI 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.

<table>
<thead>
<tr>
<th>3270</th>
<th>5250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Prerequisite Calls**

Connect Presentation Space (1)

**Call Parameters**

<table>
<thead>
<tr>
<th>Function Number</th>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data String</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>Length</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>PS Position</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>

**Return Parameters**

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The Reserve function was successful.</td>
</tr>
<tr>
<td>1</td>
<td>Your program is not connected to a host session.</td>
</tr>
<tr>
<td>5</td>
<td>Presentation space cannot be used.</td>
</tr>
<tr>
<td>9</td>
<td>A system error was encountered.</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th></th>
<th>3270</th>
<th>5250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>

**Prerequisite Calls**

There are no prerequisite calls for this function.

**Call Parameters**

<table>
<thead>
<tr>
<th></th>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Number</td>
<td>Must be 21</td>
<td></td>
</tr>
<tr>
<td>Data String</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>PS Position</td>
<td>NA</td>
<td></td>
</tr>
</tbody>
</table>

**Return Parameters**

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The Reset System function was successful.</td>
</tr>
<tr>
<td>1</td>
<td>EHLLAPI is not loaded.</td>
</tr>
<tr>
<td>9</td>
<td>A system error was encountered.</td>
</tr>
</tbody>
</table>

**Remarks**

For the PC/3270, this function can be used to check whether EHLLAPI 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.
Prerequisite Calls

Connect Presentation Space (1)

Call Parameters

<table>
<thead>
<tr>
<th>Function Number</th>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data String</td>
<td>Must be 30.</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>Length of the target data string. Overridden in EOT mode.</td>
<td></td>
</tr>
<tr>
<td>PS Position</td>
<td>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.</td>
<td></td>
</tr>
</tbody>
</table>

Return Parameters

This function returns a length and a return code.

**Length**

The following codes are defined:

<table>
<thead>
<tr>
<th>Length</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>= 0</td>
<td>The string was not found.</td>
</tr>
<tr>
<td>&gt; 0</td>
<td>The string was found at the indicated host presentation space position.</td>
</tr>
</tbody>
</table>

**Return Code**

The following codes are defined:

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The Search Field function was successful.</td>
</tr>
<tr>
<td>1</td>
<td>Your program is not connected to a host session.</td>
</tr>
<tr>
<td>2</td>
<td>Parameter error. Either the string length was zero, or EOT mode was specified but no EOT character was found in calling data string.</td>
</tr>
<tr>
<td>7</td>
<td>The host presentation space position is not valid.</td>
</tr>
<tr>
<td>9</td>
<td>A system error was encountered.</td>
</tr>
<tr>
<td>24</td>
<td>The search string was not found, or the host presentation space was unformatted.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>3270</th>
<th>5250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Prerequisite Calls**

Connect Presentation Space (1)

**Call Parameters**

<table>
<thead>
<tr>
<th></th>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Number</td>
<td>Must be 6.</td>
<td></td>
</tr>
<tr>
<td>Data String</td>
<td>Target string for search.</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>Length of the target data string. Overridden in EOT mode.</td>
<td></td>
</tr>
<tr>
<td>PS Position</td>
<td>Position within the host presentation space where the search is to begin (SRCHFRWD option) or to end (SRCHBKWD option). Overridden in SRCHALL (default) mode.</td>
<td></td>
</tr>
</tbody>
</table>

**Return Parameters**
This function returns a length and a return code.

**Length**

The following codes are defined:

<table>
<thead>
<tr>
<th>Length</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>= 0</td>
<td>The string was not found.</td>
</tr>
<tr>
<td>&gt; 0</td>
<td>The string was found at the indicated host presentation space position.</td>
</tr>
</tbody>
</table>

**Return Code**

The following codes are defined:

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The Search Presentation Space function was successful.</td>
</tr>
<tr>
<td>1</td>
<td>Your program is not connected to a host session.</td>
</tr>
<tr>
<td>2</td>
<td>An error was made in specifying parameters.</td>
</tr>
<tr>
<td>7</td>
<td>The host presentation space position is not valid.</td>
</tr>
<tr>
<td>9</td>
<td>A system error was encountered.</td>
</tr>
<tr>
<td>24</td>
<td>The search string was not found.</td>
</tr>
</tbody>
</table>

**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 specify SRCHBKWD. When this option is in effect, the search operation locates the last occurrence of the string.
3. 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.
4. The SRCHFROM option is also useful if you are looking for a keyword that might occur more than once in the host presentation space.
5. The Search Presentation Space function is useful in determining when the host presentation space is available. If your EHLLAPI 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.
6. 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 EHLLAPI is running to a host session.

<table>
<thead>
<tr>
<th>3270</th>
<th>5250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Prerequisite Calls**

There are no prerequisite calls for this function.

**Call Parameters**

<table>
<thead>
<tr>
<th>Function Number</th>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Must be 90.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data String</td>
<td>Refer to the examples.</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>Length of the target data string.</td>
<td>Overridden in EOT mode.</td>
</tr>
<tr>
<td>PS Position</td>
<td>Must be 0.</td>
<td></td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Host System</th>
<th>Common Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM/CMS</td>
<td>ASCII, JISCII, CRLF, APPEND, LRECL n, RECFM v</td>
</tr>
<tr>
<td>MVS/TSO</td>
<td>ASCII, JISCII, CRLF, APPEND, LRECL (n), RECFM (v</td>
</tr>
<tr>
<td>CICS</td>
<td>ASCII, JISCII, CRLF, BINARY, TIME n, CLEAR, NOCLEAR, PROGRESS, QUIET</td>
</tr>
<tr>
<td>i5/OS or OS/400</td>
<td>ASCII, JISCII, CRLF, APPEND, SRC, LRECL n, TIME n, CLEAR, NOCLEAR, PROGRESS, QUIET</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>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.</td>
</tr>
<tr>
<td>3</td>
<td>File transfer complete.</td>
</tr>
<tr>
<td>4</td>
<td>File transfer complete with segmented records.</td>
</tr>
<tr>
<td>5</td>
<td>Workstation file name is not valid or not found. File transfer was canceled.</td>
</tr>
<tr>
<td>9</td>
<td>A system error was encountered.</td>
</tr>
<tr>
<td>27</td>
<td>File transfer terminated because of either a Cancel button or the timeout set by the Set Session Parameter (9) function.</td>
</tr>
<tr>
<td>101</td>
<td>File transfer was successful (transfer to/from CICS).</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>3270</th>
<th>5250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Prerequisite Calls

Connect Presentation Space (1)

Call Parameters

<table>
<thead>
<tr>
<th></th>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Number</td>
<td>Must be 3.</td>
<td></td>
</tr>
<tr>
<td>Data String</td>
<td>A string of keystrokes, maximum 255.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Uppercase and lowercase ASCII characters are represented</td>
<td></td>
</tr>
<tr>
<td></td>
<td>literally.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Function keys and shifted function keys are represented</td>
<td></td>
</tr>
<tr>
<td></td>
<td>by mnemonics.</td>
<td></td>
</tr>
</tbody>
</table>
See Keyboard Mnemonics.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>Length of the source data string.</td>
</tr>
<tr>
<td></td>
<td>Overridden if in EOT mode.</td>
</tr>
<tr>
<td>PS Position</td>
<td>NA</td>
</tr>
</tbody>
</table>

**Return Parameters**

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The keystrokes were sent; status is normal.</td>
</tr>
<tr>
<td>1</td>
<td>Your program is not connected to a host session.</td>
</tr>
<tr>
<td>2</td>
<td>An incorrect parameter was passed to EHLLAPI.</td>
</tr>
<tr>
<td>4</td>
<td>The host session was busy; all of the keystrokes could not be sent.</td>
</tr>
<tr>
<td>5</td>
<td>Input to the target session was inhibited or rejected; all of the keystrokes could not be sent.</td>
</tr>
<tr>
<td>9</td>
<td>A system error was encountered.</td>
</tr>
</tbody>
</table>

**Remarks**

1. 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.
2. 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.
3. If the host is busy, input might be rejected.
4. 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.
5. 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.
6. 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.
7. 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.

<table>
<thead>
<tr>
<th>3270</th>
<th>5250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Prerequisite Calls**

Connect Presentation Space (1)

**Call Parameters**

<table>
<thead>
<tr>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Number</td>
<td>Must be 40</td>
</tr>
<tr>
<td>Data String</td>
<td>NA</td>
</tr>
<tr>
<td>Length</td>
<td>NA</td>
</tr>
<tr>
<td>PS Position</td>
<td>Desired cursor position in the connected host presentation space</td>
</tr>
</tbody>
</table>

**Return Parameters**

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Cursor was successfully located at the specified position.</td>
</tr>
<tr>
<td>1</td>
<td>Your program is not connected to a host session.</td>
</tr>
<tr>
<td>4</td>
<td>The session is busy.</td>
</tr>
<tr>
<td>7</td>
<td>A cursor location less than 1 or greater than the size of the connected host presentation space was specified.</td>
</tr>
<tr>
<td>9</td>
<td>A system error occurred.</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Function Number</th>
<th>Function Name</th>
<th>See Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Connect Presentation Space</td>
<td>11, 23, 24</td>
</tr>
<tr>
<td>3</td>
<td>Send Key</td>
<td>1, 2, 9, 10, 19</td>
</tr>
<tr>
<td>4</td>
<td>Wait</td>
<td>12</td>
</tr>
<tr>
<td>5</td>
<td>Copy Presentation Space</td>
<td>5, 13, 14, 15, 17, 20, 21, 22</td>
</tr>
<tr>
<td>6</td>
<td>Search Presentation Space</td>
<td>1, 2, 3, 4</td>
</tr>
<tr>
<td>8</td>
<td>Copy Presentation Space to String</td>
<td>5, 13, 14, 15, 17, 20, 21, 22</td>
</tr>
<tr>
<td>10</td>
<td>Query Sessions</td>
<td>16, 22</td>
</tr>
<tr>
<td>15</td>
<td>Copy String to Presentation Space</td>
<td>1, 2, 13, 14, 18, 20, 21, 22</td>
</tr>
<tr>
<td>18</td>
<td>Pause</td>
<td>6</td>
</tr>
<tr>
<td>30</td>
<td>Search Field</td>
<td>1, 2, 3, 4, 22</td>
</tr>
<tr>
<td>33</td>
<td>Copy String to Field</td>
<td>1, 2, 13, 14, 18, 20, 21, 22</td>
</tr>
<tr>
<td>34</td>
<td>Copy Field to String</td>
<td>5, 13, 14, 17, 20, 21, 22</td>
</tr>
<tr>
<td>51</td>
<td>Get Key</td>
<td>9, 12</td>
</tr>
<tr>
<td>90</td>
<td>Send File</td>
<td>1, 2, 7, 8</td>
</tr>
<tr>
<td>91</td>
<td>Receive File</td>
<td>1, 2, 7, 8</td>
</tr>
<tr>
<td>101</td>
<td>Connect Window Services</td>
<td>23, 24</td>
</tr>
</tbody>
</table>

Note: Items 20 and 21 in this table are for DBCS only

Prerequisite Calls

There are no prerequisite calls for this function.

Call Parameters

<table>
<thead>
<tr>
<th>Function Number</th>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data String</td>
<td>String containing the desired values of those session options that are to be</td>
<td></td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>Explicit length of the source data string (the STREOT option is not allowed).</td>
</tr>
<tr>
<td>PS Position</td>
<td>NA.</td>
</tr>
</tbody>
</table>

### Session Options

The following tables show the session options. The default is underlined.

1. 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)**.

<table>
<thead>
<tr>
<th>Value</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>STRLEN</td>
<td>An explicit length is passed for all strings.</td>
</tr>
<tr>
<td>STREOT</td>
<td>Lengths are not explicitly coded. Calling (source) data strings are terminated with an EOT character.</td>
</tr>
</tbody>
</table>

2. 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 EHLLAPI 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)**.

<table>
<thead>
<tr>
<th>Value</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>EOT=c</td>
<td>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.</td>
</tr>
</tbody>
</table>

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.

3. The values in the following table affect the **Search Presentation Space (6)** and **Search Field (30)** search functions.

<table>
<thead>
<tr>
<th>Value</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRCHALL</td>
<td>The <strong>Search Presentation Space (6)</strong> function and <strong>Search Field (30)</strong> function scan the entire host presentation space or field.</td>
</tr>
<tr>
<td>SRCHFROM</td>
<td>The <strong>Search Presentation Space (6)</strong> function and <strong>Search Field (30)</strong> function start from a specified PS position (for SRCHFRWD) or end at a specified PS position (for SRCHBKWD).</td>
</tr>
</tbody>
</table>

4. 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.

<table>
<thead>
<tr>
<th>Value</th>
<th>Explanation</th>
</tr>
</thead>
</table>
The Search Presentation Space (6) function and Search Field (30) function perform in an ascending direction.

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.

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).

<table>
<thead>
<tr>
<th>Value</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOATTRB</td>
<td>Convert all unknown values to blanks.</td>
</tr>
<tr>
<td>ATTRB</td>
<td>Pass back all codes that do not have an ASCII equivalent as their original values.</td>
</tr>
<tr>
<td>NULLATTRB</td>
<td>Convert all field attributes to null characters.</td>
</tr>
</tbody>
</table>

The values in the following table affect the Pause (18) function.

<table>
<thead>
<tr>
<th>Value</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>FPAUSE</td>
<td>A full-duration pause lasts for however long you specified in the Pause (18) function.</td>
</tr>
<tr>
<td>IPAUSE</td>
<td>Interruptible pause. After the Start Host Notification (23) function is executed, a host event satisfies a pause.</td>
</tr>
</tbody>
</table>

The values in the following table determine whether messages generated by file transfer functions Send File (90) and Receive File (91) are displayed.

<table>
<thead>
<tr>
<th>Value</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOQUIET</td>
<td>SEND and RECEIVE messages are displayed.</td>
</tr>
<tr>
<td>QUIET</td>
<td>SEND and RECEIVE messages are not displayed.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Value</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIMEOUT=0</td>
<td>A Cancel is automatically issued following a 20-second (approximate) delay.</td>
</tr>
<tr>
<td>TIMEOUT=c</td>
<td>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)</td>
</tr>
<tr>
<td>1 0.5</td>
<td>2 1.0</td>
</tr>
<tr>
<td>3 1.5</td>
<td>4 2.0</td>
</tr>
<tr>
<td>5 2.5</td>
<td>6 3.0</td>
</tr>
<tr>
<td>7 3.5</td>
<td>8 4.0</td>
</tr>
<tr>
<td>9 4.5</td>
<td>J 5.0</td>
</tr>
<tr>
<td>K 5.5</td>
<td>L 6.0</td>
</tr>
<tr>
<td>M 6.5</td>
<td>N 7.0</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Value</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESC=c</td>
<td>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.</td>
</tr>
</tbody>
</table>

10. The values in the following table determine whether EHLLAPI automatically precedes strings sent using the **Send Key (3)** function with a reset.

<table>
<thead>
<tr>
<th>Value</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTORESET</td>
<td>EHLLAPI attempts to reset all inhibited conditions by prefixing all strings of keys sent using the <strong>Send Key (3)</strong> function with a reset.</td>
</tr>
<tr>
<td>NORESET</td>
<td>Do not AUTORESET.</td>
</tr>
</tbody>
</table>

11. The values in the following table affect the manner in which the **Connect Presentation Space (1)** command function.

<table>
<thead>
<tr>
<th>Value</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONLOG</td>
<td>Establishes a logical connection between the workstation session and a host session. During Connect, does not jump to the requested presentation space.</td>
</tr>
<tr>
<td>CONPHYS</td>
<td>Establishes a physical connection between the workstation session and a host session. During Connect, jumps to the requested presentation space.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Value</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>TWAIT</td>
<td>For the <strong>Wait (4)</strong> function, waits up to a minute before timing out on XCLOCK (X[]) or XSYSTEM. For the <strong>Get Key (51)</strong> function, does not return control to your EHLLAPI application program until it has intercepted a key (normal or AID key based on the option specified under the Start Keystroke Intercept (50) function).</td>
</tr>
<tr>
<td>LWAIT</td>
<td>For the <strong>Wait (4)</strong> 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 <strong>Get Key (51)</strong> function, does not return control to your EHLLAPI application program until it has intercepted a key (normal or AID key based on the option specified under the Start Keystroke Intercept (50) function).</td>
</tr>
<tr>
<td>NWAIT</td>
<td>For the <strong>Wait (4)</strong> function, checks status and returns immediately (no wait). For the <strong>Get Key (51)</strong> function, returns return code 25 (keystrokes not available) in the fourth parameter if nothing is queued matching the option specified under the <strong>Start Keystroke Intercept (50)</strong> function.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Value</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOEAB</td>
<td>Pass data only, no EABs.</td>
</tr>
<tr>
<td>EAB</td>
<td>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 x 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...</td>
</tr>
</tbody>
</table>
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).

<table>
<thead>
<tr>
<th>Value</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOXLATE</td>
<td>EABs are not translated.</td>
</tr>
<tr>
<td>XLATE</td>
<td>EABs are translated to the PC color graphics adapter (CGA) format.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Value</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLANK</td>
<td>Convert all unknown values to X'20'.</td>
</tr>
<tr>
<td>NOBLANK</td>
<td>Convert all unknown values to X'00'.</td>
</tr>
</tbody>
</table>

16. The values in the following table affect the presentation space size that is
returned by the Query Sessions (10).

<table>
<thead>
<tr>
<th>Value</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFGSIZE</td>
<td>Returns the configured size of the connected presentation space. This option ignores any override of the configured size by the host.</td>
</tr>
<tr>
<td>NOCFG SIZE</td>
<td>Returns the current size of the connected presentation space.</td>
</tr>
</tbody>
</table>

17. The values in the following table affect Copy Presentation Space (5), Copy
Presentation Space to String (8), and Copy Field to String (34).

<table>
<thead>
<tr>
<th>Value</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISPLAY</td>
<td>Copy nondisplay fields in the presentation space to the target buffer area in the same manner as display fields. Current applications function normally.</td>
</tr>
<tr>
<td>NODISPLAY</td>
<td>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 window without displaying confidential information, such as passwords.</td>
</tr>
</tbody>
</table>

18. The values in the following table affect Copy String to Presentation Space (15)
and Copy String to Field (33).

<table>
<thead>
<tr>
<th>Value</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOPUTEAB</td>
<td>EAB (or EAD for DBCS) is not contained in the data string of Copy String to Presentation Space or</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Value</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RETRY</td>
<td>Continues to attempt to send keystrokes until they are sent or until a 4-minute timeout occurs.</td>
</tr>
<tr>
<td>NORETRY</td>
<td>Returns immediately after determining the keyboard is blocked or in use.</td>
</tr>
</tbody>
</table>

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)**.

<table>
<thead>
<tr>
<th>Value</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOEAD</td>
<td>DBCS attribute characters are not passed.</td>
</tr>
<tr>
<td>EAD</td>
<td>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.)</td>
</tr>
</tbody>
</table>

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)**.

<table>
<thead>
<tr>
<th>Value</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOSO</td>
<td>Pass the presentation space data except Shift-in (SI) and Shift-out (SO) control characters.</td>
</tr>
<tr>
<td>SO</td>
<td>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.</td>
</tr>
<tr>
<td>SPACESO</td>
<td>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.</td>
</tr>
</tbody>
</table>

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)**.

<table>
<thead>
<tr>
<th>Value</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXTEND_PS</td>
<td>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.</td>
</tr>
<tr>
<td>NOEXTEND_PS</td>
<td>The presentation space is not extended when the above condition occurs. This is the default value.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Value</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUPER_WRITE</td>
<td>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</td>
</tr>
</tbody>
</table>
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).

<table>
<thead>
<tr>
<th>Value</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOKEY</td>
<td>Allows the application to be compatible with existing applications that do not specify the KEY parameter.</td>
</tr>
<tr>
<td>$\text{KEY}$nnnnnnnn</td>
<td>Uses a keyword to restrict sharing access to the presentation space that it supports. The keyword must be exactly 8 bytes in length.</td>
</tr>
</tbody>
</table>

**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:

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The session parameters have been set.</td>
</tr>
<tr>
<td>2</td>
<td>One or more parameters were not valid.</td>
</tr>
<tr>
<td>9</td>
<td>A system error was encountered.</td>
</tr>
</tbody>
</table>

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.
Prerequisite Calls

There are no prerequisite calls for this function.

Call Parameters

<table>
<thead>
<tr>
<th>Byte</th>
<th>Definition</th>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Function Number</td>
<td>Must be 41</td>
<td></td>
</tr>
<tr>
<td>Data String</td>
<td></td>
<td>See the following table</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td></td>
<td>5 or 6</td>
<td>Must be 12</td>
</tr>
<tr>
<td>PS Position</td>
<td></td>
<td>NA</td>
<td></td>
</tr>
</tbody>
</table>

The data string contains the following items.

<table>
<thead>
<tr>
<th>Byte</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Standard Enhanced</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2-4</td>
<td>Reserved.</td>
</tr>
<tr>
<td>4-5</td>
<td>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.</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>6-8</td>
<td>Reserved.</td>
</tr>
<tr>
<td>2-3</td>
<td>9-12</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Byte</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Standard Enhanced</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2-8</td>
<td>Reserved.</td>
</tr>
<tr>
<td>9-12</td>
<td>4 byte value in which the event object address is returned by EHLLAPI. The application can wait for this event object. (32-bit only).</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Byte</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Standard</td>
</tr>
<tr>
<td></td>
<td>Enhanced</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>A 1-character presentation space short name (PSID)</td>
</tr>
<tr>
<td>2-8</td>
<td>Reserved</td>
</tr>
<tr>
<td>2-3</td>
<td>9-10</td>
</tr>
<tr>
<td></td>
<td>Task ID of asynchronous message mode</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The Start Close Intercept function was successful.</td>
</tr>
<tr>
<td>1</td>
<td>An incorrect host presentation space was specified.</td>
</tr>
<tr>
<td>2</td>
<td>A parameter error occurred.</td>
</tr>
<tr>
<td>9</td>
<td>A system error occurred.</td>
</tr>
<tr>
<td>10</td>
<td>The function is not supported by the emulation program.</td>
</tr>
</tbody>
</table>

**Remarks**

1. The returned event object or semaphore is in a non-signaled state when the start request function returns. 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.

<table>
<thead>
<tr>
<th>3270</th>
<th>5250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Prerequisite Calls**

There are no prerequisite calls for this function.

**Call Parameters**

<table>
<thead>
<tr>
<th>Function Number</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Must be 80</td>
</tr>
<tr>
<td>Data String</td>
<td>Preallocated structure; see the following table</td>
</tr>
<tr>
<td>Length</td>
<td>16</td>
</tr>
<tr>
<td>PSPosition</td>
<td>NA</td>
</tr>
</tbody>
</table>

The calling data structure contains these elements

<table>
<thead>
<tr>
<th>Byte</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A 1-character presentation space short-name (PSID).</td>
</tr>
<tr>
<td>2-4</td>
<td>Reserved</td>
</tr>
<tr>
<td>5</td>
<td>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.</td>
</tr>
<tr>
<td>6-8</td>
<td>Reserved</td>
</tr>
<tr>
<td>9-12</td>
<td>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).</td>
</tr>
<tr>
<td>13</td>
<td>This should contain the character C if position 5 is A or M.</td>
</tr>
<tr>
<td>14-16</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

**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:

<table>
<thead>
<tr>
<th>Byte</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A 1-character presentation space short-name (PSID)</td>
</tr>
<tr>
<td>2-8</td>
<td>Reserved</td>
</tr>
<tr>
<td>9-12</td>
<td>4-byte binary value in which the event object handle is returned by EHLAPI. The application can wait for this event object.</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Byte</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A 1-character presentation space short-name (PSID)</td>
</tr>
</tbody>
</table>
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**

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The function was successful</td>
</tr>
<tr>
<td>1</td>
<td>An incorrect PSID was specified</td>
</tr>
<tr>
<td>2</td>
<td>An error was made in designating parameters</td>
</tr>
<tr>
<td>9</td>
<td>A system error was encountered</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>3270</th>
<th>5250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Prerequisite Calls**

There are no prerequisite calls for this function.
Call Parameters

<table>
<thead>
<tr>
<th></th>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Number</td>
<td>Must be 23</td>
<td></td>
</tr>
<tr>
<td>Data String</td>
<td>Preallocated string; see the following table</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>6 or 7 implied</td>
<td>16</td>
</tr>
<tr>
<td>PS Position</td>
<td>NA</td>
<td></td>
</tr>
</tbody>
</table>

The calling data string contains these elements:

<table>
<thead>
<tr>
<th>Byte</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Standard</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>2-4</td>
<td>Reserved.</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>6-8</td>
<td>Reserved.</td>
</tr>
<tr>
<td>3-4</td>
<td>9-12</td>
</tr>
<tr>
<td>7</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Byte</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Standard</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Reserved.

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 (2 for standard interface) and the function is completed successfully, the following data string is returned:

<table>
<thead>
<tr>
<th>Byte</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A 1-character presentation space short name (PSID)</td>
</tr>
<tr>
<td>2-8</td>
<td>Reserved</td>
</tr>
<tr>
<td>3-4</td>
<td>Task ID of asynchronous message mode</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The Start Host Notification function was successful.</td>
</tr>
<tr>
<td>1</td>
<td>An incorrect host presentation space was specified.</td>
</tr>
<tr>
<td>2</td>
<td>An error was made in designating parameters.</td>
</tr>
<tr>
<td>9</td>
<td>A system error was encountered.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Function Number</th>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data String</td>
<td>Must be 50</td>
<td>See the following table</td>
</tr>
<tr>
<td>Length</td>
<td>Keystroke buffer size EHLLAPI allocates 32 bytes minimum for this buffer.</td>
<td></td>
</tr>
<tr>
<td>PS Position</td>
<td>NA</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Byte</th>
<th>Definition</th>
</tr>
</thead>
</table>
| 1    | One of the following values:  
|      | • 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. |
| 5    | An option code character:  
|      | • 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. |
| 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). |
| 13   | One of the following values if position 5 (2 for 16-bit) is M:  
|      | • D for AID keystrokes only.  
|      | • L for all keystrokes. |

Prerequisite Calls

There are no prerequisite calls for this function.

Call Parameters
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:

<table>
<thead>
<tr>
<th>Byte</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td>Enhanced</td>
</tr>
<tr>
<td>1</td>
<td>A 1-character presentation space short name (PSID)</td>
</tr>
<tr>
<td>2-8</td>
<td>Reserved</td>
</tr>
<tr>
<td>3-4</td>
<td>Task ID of asynchronous message mode</td>
</tr>
</tbody>
</table>

Note:
If a user sends keystrokes to a session, 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 parameter contains return code 0, which shows that the function was successful, and LOWORD of lParam parameter contains function number 50.

Return Parameters

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The Start Keystroke Intercept function was successful.</td>
</tr>
<tr>
<td>1</td>
<td>An incorrect presentation space was specified.</td>
</tr>
<tr>
<td>2</td>
<td>An incorrect option was specified.</td>
</tr>
<tr>
<td>4</td>
<td>The execution of the function was inhibited because the target presentation space was busy.</td>
</tr>
<tr>
<td>9</td>
<td>A system error was encountered. Release is being used.</td>
</tr>
</tbody>
</table>

Remarks

1. 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.
2. 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.
3. 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.

<table>
<thead>
<tr>
<th>3270</th>
<th>5250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Prerequisite Calls**

Connect Presentation Space (1)

**Call Parameters**

<table>
<thead>
<tr>
<th>Standard Interface</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Number</td>
<td>Must be 110</td>
</tr>
<tr>
<td>Data String</td>
<td>See the following table</td>
</tr>
<tr>
<td>Length</td>
<td>Length of macro name, plus 3</td>
</tr>
<tr>
<td>PS Position</td>
<td>NA</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Byte</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-2</td>
<td>Reserved</td>
</tr>
<tr>
<td>3-n</td>
<td>Null terminated macro name</td>
</tr>
</tbody>
</table>

**Return Parameters**

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The Start Playing Macro function was successful.</td>
</tr>
<tr>
<td>1</td>
<td>The programs is not connected to a host session.</td>
</tr>
<tr>
<td>2</td>
<td>An error was made in specifying parameters.</td>
</tr>
<tr>
<td>9</td>
<td>A system error was encountered.</td>
</tr>
</tbody>
</table>

5.2.1.47 Stop Close Intercept (43)

The Stop Close Intercept function allows the application to turn off the Stop 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.

<table>
<thead>
<tr>
<th>3270</th>
<th>5250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Prerequisite Calls**

Start Close Intercept (41)
Call Parameters

<table>
<thead>
<tr>
<th></th>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Number</td>
<td>Must be 43</td>
<td>Must be 4</td>
</tr>
<tr>
<td>Data String</td>
<td>1-character short session ID of the host presentation space</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>1</td>
<td>Must be 4</td>
</tr>
<tr>
<td>PS Position</td>
<td>NA</td>
<td></td>
</tr>
</tbody>
</table>

The calling data string can contain:

<table>
<thead>
<tr>
<th>Byte</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A 1-character presentation space short name (PSID)</td>
</tr>
<tr>
<td>2-4</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

Return Parameters

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The Stop Close Intercept function was successful.</td>
</tr>
<tr>
<td>1</td>
<td>An incorrect host presentation space was specified.</td>
</tr>
<tr>
<td>2</td>
<td>An error was made in specifying parameters.</td>
</tr>
<tr>
<td>8</td>
<td>No previous Start Close Intercept (41) function was issued.</td>
</tr>
<tr>
<td>9</td>
<td>A system error occurred.</td>
</tr>
<tr>
<td>12</td>
<td>The session stopped.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th></th>
<th>3270</th>
<th>5250</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Prerequisite Calls

Start Communication Notification (80)

Call Parameters

<table>
<thead>
<tr>
<th></th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Number</td>
<td>Must be 82</td>
</tr>
<tr>
<td>Data String</td>
<td>1-character short name of the host presentation space, or a blank or null indicating request for updates to the host-connected</td>
</tr>
</tbody>
</table>
The calling data structure contains these elements:

<table>
<thead>
<tr>
<th>Byte</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A 1-character presentation space short name (PSID)</td>
</tr>
<tr>
<td>2-4</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

**Return Parameters**

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The function was successful</td>
</tr>
<tr>
<td>1</td>
<td>An incorrect PSID was specified</td>
</tr>
<tr>
<td>8</td>
<td>No prior call to Start Communication Notification (80) function was called for the PSID</td>
</tr>
<tr>
<td>9</td>
<td>A system error was encountered</td>
</tr>
</tbody>
</table>

### 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.

<table>
<thead>
<tr>
<th>3270</th>
<th>5250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Prerequisite Calls**

**Start Host Notification (23)**

**Call Parameters**

<table>
<thead>
<tr>
<th></th>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Number</td>
<td>Must be 121</td>
<td></td>
</tr>
<tr>
<td>Data String</td>
<td>See the following note</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>1 is implied</td>
<td>Must be 4</td>
</tr>
<tr>
<td>PS Position</td>
<td>NA</td>
<td></td>
</tr>
</tbody>
</table>

The calling data string can contain:

<table>
<thead>
<tr>
<th>Byte</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2-4</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

**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**

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The Stop Host Notification function was successful.</td>
</tr>
<tr>
<td>1</td>
<td>An incorrect host presentation space was specified.</td>
</tr>
<tr>
<td>8</td>
<td>No previous Start Host Notification (23) function issued.</td>
</tr>
<tr>
<td>9</td>
<td>A system error was encountered.</td>
</tr>
</tbody>
</table>

### 5.2.1.50 Stop Keystroke Intercept (53)

The Stop Keystroke Intercept function ends your application program's ability to intercept keystrokes.

<table>
<thead>
<tr>
<th>3270</th>
<th>5250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Prerequisite Calls**

Start Keystroke Intercept (50)

**Call Parameters**

<table>
<thead>
<tr>
<th></th>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Number</td>
<td>Must be 53</td>
<td></td>
</tr>
<tr>
<td>Data String</td>
<td>Short name of the target presentation space (PSID)</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>1 is implied</td>
<td>Must be 4</td>
</tr>
<tr>
<td>PS Position</td>
<td>NA</td>
<td></td>
</tr>
</tbody>
</table>

The calling data string can contain:

<table>
<thead>
<tr>
<th>Byte</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A 1-character presentation space short name (PSID)</td>
</tr>
<tr>
<td>2-4</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

**Return Parameters**

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The Stop Keystroke Intercept function was successful.</td>
</tr>
<tr>
<td>1</td>
<td>An incorrect presentation space was specified.</td>
</tr>
<tr>
<td>8</td>
<td>No prior Start Keystroke Intercept (50) function called for this presentation space.</td>
</tr>
<tr>
<td>9</td>
<td>A system error was encountered.</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>3270</th>
<th>5250</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Prerequisite Calls**

Connect Presentation Space (1)

**Call Parameters**

<table>
<thead>
<tr>
<th></th>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Number</td>
<td>Must be 4</td>
<td></td>
</tr>
<tr>
<td>Data String</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>PS Position</td>
<td>NA</td>
<td></td>
</tr>
</tbody>
</table>

**Return Parameters**

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The keyboard is unlocked and ready for input.</td>
</tr>
<tr>
<td>1</td>
<td>Your application program is not connected to a valid session.</td>
</tr>
<tr>
<td>4</td>
<td>Timeout while still in XCLOCK (X [ ]) or XSYSTEM.</td>
</tr>
<tr>
<td>5</td>
<td>The keyboard is locked.</td>
</tr>
<tr>
<td>9</td>
<td>A system error was encountered.</td>
</tr>
</tbody>
</table>

**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.
### Prerequisite Calls

**Connect Window Services (101)**

### Call Parameters

<table>
<thead>
<tr>
<th></th>
<th>Standard Interface</th>
<th>Enhanced Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Function Number</strong></td>
<td>Must be 104</td>
<td></td>
</tr>
<tr>
<td><strong>Data String</strong></td>
<td>See the following table</td>
<td></td>
</tr>
<tr>
<td><strong>Length</strong></td>
<td>16 or 20</td>
<td>24 or 28</td>
</tr>
<tr>
<td><strong>PS Position</strong></td>
<td>NA</td>
<td></td>
</tr>
</tbody>
</table>

The calling data string can contain:

<table>
<thead>
<tr>
<th><strong>Byte</strong></th>
<th><strong>Standard</strong></th>
<th><strong>Definition</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>A 1-character presentation space short name (PSID)</td>
</tr>
<tr>
<td>2-4</td>
<td>5</td>
<td>A request option value, select one of the following values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• X'01' for set status</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• X'02' for query for status</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• X'03' for query for extended status</td>
</tr>
<tr>
<td>6</td>
<td>7-8</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

If the request option value is X'01' (set status):

<table>
<thead>
<tr>
<th><strong>Byte</strong></th>
<th><strong>Definition</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>3-4</td>
<td>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:</td>
</tr>
<tr>
<td></td>
<td>• X'0001' Change the window size. (Not valid with minimize, maximize, restore, or move.)</td>
</tr>
<tr>
<td></td>
<td>• X'0002' Move the window. (Not valid with minimize, maximize, size, or restore.) X'0004' ZORDER window replacement.</td>
</tr>
<tr>
<td></td>
<td>• X'0008' Set the window to visible.</td>
</tr>
<tr>
<td></td>
<td>• X'0010' Set the window to invisible.</td>
</tr>
<tr>
<td></td>
<td>• 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.)</td>
</tr>
<tr>
<td></td>
<td>• 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.)</td>
</tr>
<tr>
<td></td>
<td>• X'0400' Set the window to minimized. (Not valid with maximize, restore, size, or move.)</td>
</tr>
<tr>
<td></td>
<td>• X'0800' Set the window to maximized. (Not valid with minimize, maximize, size, or move.)</td>
</tr>
<tr>
<td></td>
<td>• X'1000' Restore the window. (Not valid with minimize, maximize, size, or move.)</td>
</tr>
</tbody>
</table>
A 16- or 32-bit word containing the X window position coordinate. (Ignored if the move option is not set.)

A 16- or 32-bit word containing the Y window position coordinate. (Ignored if the move option is not set.)

A 16- or 32-bit word containing the X window size in device units. (Ignored if the size option is not set.)

A 16- or 32-bit word containing the Y window size in device units. (Ignored if the size option is not set.)

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:
- 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):

<table>
<thead>
<tr>
<th>Byte</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-4</td>
<td>7-8</td>
</tr>
</tbody>
</table>
|      | 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.
|      | 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.  |

A 16- or 32-bit word containing the X window position coordinate. (Ignored if the move option is not set.)

A 16- or 32-bit word containing the Y window position coordinate. (Ignored if the move option is not set.)

A 16- or 32-bit word containing the X window size in device units. (Ignored if the size option is not set.)

A 16- or 32-bit word containing the Y window size in device units. (Ignored if the size option is not set.)

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:
- 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):

<table>
<thead>
<tr>
<th>Byte</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-4</td>
<td>7-8</td>
</tr>
</tbody>
</table>
|      | 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.
|      | 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.  |

A 16- or 32-bit word containing the current font size in the X-dimension. The value is in screen pels.

A 16- or 32-bit word containing the current font size in the Y-dimension. The value is in screen pels.
Return Parameters

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The Window Status function was successful.</td>
</tr>
<tr>
<td>1</td>
<td>The presentation space was not valid or not connected.</td>
</tr>
<tr>
<td>2</td>
<td>An incorrect option was specified.</td>
</tr>
<tr>
<td>9</td>
<td>A system error occurred.</td>
</tr>
<tr>
<td>12</td>
<td>The session stopped.</td>
</tr>
</tbody>
</table>

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 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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data String</td>
<td>RECEIVE command parameters.</td>
</tr>
<tr>
<td>Data Length</td>
<td>Length of Data String. NA if session option EOT is specified.</td>
</tr>
</tbody>
</table>
### Return Codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WHLLOK</td>
<td>File transfer started successfully.</td>
</tr>
<tr>
<td>WHLLPARAMETERERROR</td>
<td>Parameter error or Data Length is zero or greater than 255.</td>
</tr>
<tr>
<td>WHLLFTXCOMPLETE</td>
<td>File transfer complete.</td>
</tr>
<tr>
<td>WHLLFTXSEGMENTED</td>
<td>Transfer is complete with segmented records.</td>
</tr>
<tr>
<td>WHLLSYSERROR</td>
<td>The function failed due to a system error.</td>
</tr>
<tr>
<td>WHLLTRANSABORTED</td>
<td>File transfer aborted, either due to the user clicking the cancel button or because the timeout period has elapsed.</td>
</tr>
<tr>
<td>WHLLFILENOTFOUND</td>
<td>PC file not found.</td>
</tr>
<tr>
<td>WHLLFTXCOMPLETECICS</td>
<td>File transfer was successful (transfer to CICS).</td>
</tr>
<tr>
<td>WHLLACCESSDENIED</td>
<td>Access denied to PC file.</td>
</tr>
<tr>
<td>WHLLMEMORY</td>
<td>Insufficient memory.</td>
</tr>
<tr>
<td>WHLLUNVALIDENVIRONMENT</td>
<td>Invalid environment.</td>
</tr>
</tbody>
</table>

### 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.

**IParm**

If the low byte of wParm is zero (in progress), IParm is the number of bytes transferred. If the low byte wParm is one (completed), IParm 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, lpwReturnValue)

Call Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data String</td>
<td>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.</td>
</tr>
<tr>
<td>Data Length</td>
<td>Must be specified.</td>
</tr>
<tr>
<td>PS Position</td>
<td>NA</td>
</tr>
</tbody>
</table>

Return Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data String</td>
<td>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.</td>
</tr>
</tbody>
</table>

Return Code

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WHLOK</td>
<td>The function was successful.</td>
</tr>
<tr>
<td>WHLNOTCONNECTED</td>
<td>An invalid presentation space was specified.</td>
</tr>
<tr>
<td>WHLPARAMETERERROR</td>
<td>An invalid option was specified.</td>
</tr>
<tr>
<td>WHLLSYERROR</td>
<td>The function failed due to a system error.</td>
</tr>
<tr>
<td>WHLLCANCEL</td>
<td>The asynchronous function was cancelled.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data String</td>
<td>SEND command parameters.</td>
</tr>
<tr>
<td>Data Length</td>
<td>Length of Data String. NA if session option EOT is specified.</td>
</tr>
<tr>
<td>PS Position</td>
<td>NA</td>
</tr>
</tbody>
</table>

Return Codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WHLLOK</td>
<td>File transfer started successfully.</td>
</tr>
<tr>
<td>WHLLPARAMETERERROR</td>
<td>Parameter error or Data Length is zero or greater than 255.</td>
</tr>
<tr>
<td>WHLLFTXCOMPLETE</td>
<td>File transfer complete.</td>
</tr>
<tr>
<td>WHLLFTXSEGMENTED</td>
<td>Transfer is complete with segmented records.</td>
</tr>
<tr>
<td>WHLLSYSERROR</td>
<td>The function failed due to a system error.</td>
</tr>
<tr>
<td>WHLLTRANSABORTED</td>
<td>File transfer aborted, either due to the user clicking the cancel button or because the timeout period has elapsed.</td>
</tr>
<tr>
<td>WHLLFILENOTFOUND</td>
<td>PC file not found.</td>
</tr>
<tr>
<td>WHLLFTXCOMPLETECICS</td>
<td>File transfer was successful (transfer to CICS).</td>
</tr>
<tr>
<td>WHLLACCESS DENIED</td>
<td>Access denied to PC file.</td>
</tr>
<tr>
<td>WHLLMEMORY</td>
<td>Insufficient memory.</td>
</tr>
<tr>
<td>WHLLINVALIDENVIRONMENT</td>
<td>Invalid environment.</td>
</tr>
</tbody>
</table>

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.

\( w\text{Parm} \)

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.

\( i\text{Parm} \)

If the low byte of wParm is zero (in progress), iParm is the number of bytes transferred. If the low byte wParm is one (completed), iParm 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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data String</td>
<td>A 7-byte string in the following format:</td>
</tr>
<tr>
<td></td>
<td>• Byte 1  Short name session ID of the desired Host session, or space or null for the current Host session.</td>
</tr>
<tr>
<td></td>
<td>• Byte 2  Notification mode. &quot;P&quot; for presentation space update only, &quot;O&quot; for OIA update only, &quot;B&quot; for both presentation space and OIA updates. When calling WinHLLAPIAsync, this position can be &quot;A&quot;.</td>
</tr>
<tr>
<td></td>
<td>• Byte 3-6 Not used. Provided for compatibility with older applications.</td>
</tr>
<tr>
<td></td>
<td>• 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.</td>
</tr>
<tr>
<td>Data Length</td>
<td>Length of Host event buffer (256 recommended).</td>
</tr>
<tr>
<td>PS Position</td>
<td>NA</td>
</tr>
</tbody>
</table>

Return Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data String</td>
<td>Same as Data String on the call.</td>
</tr>
</tbody>
</table>

Return Codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WHLLOK</td>
<td>Host notification enabled.</td>
</tr>
<tr>
<td>WHILLNOTCONNECTED</td>
<td>The specified Host session is invalid.</td>
</tr>
<tr>
<td>WHLLPARAMETERERROR</td>
<td>One of more parameters are invalid.</td>
</tr>
<tr>
<td>WHLLSYSERROR</td>
<td>The function failed due to a system error.</td>
</tr>
<tr>
<td>WHLLCANCEL</td>
<td>The asynchronous function was cancelled.</td>
</tr>
</tbody>
</table>

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**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data String</td>
<td>A 6-byte string in the following format:</td>
</tr>
<tr>
<td></td>
<td>● Byte 1  Session short name, or space or null for the current Host session.</td>
</tr>
<tr>
<td></td>
<td>● Byte 2  Keystroke intercept code. &quot;D&quot; causes only AID keystrokes to be</td>
</tr>
<tr>
<td></td>
<td>intercepted; &quot;L&quot; causes all keystrokes to be intercepted.</td>
</tr>
<tr>
<td></td>
<td>● Bytes 3-6  Reserved</td>
</tr>
<tr>
<td>Data Length</td>
<td>Variable (256 is recommended)</td>
</tr>
<tr>
<td>PS Position</td>
<td>NA</td>
</tr>
</tbody>
</table>

**Return Code**

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WHLLOK</td>
<td>Keystroke intercept has been initiated.</td>
</tr>
<tr>
<td>WHLNOTCONNECTED</td>
<td>The Host session presentation space is invalid.</td>
</tr>
<tr>
<td>WHLPPARAMETERERROR</td>
<td>One or more parameters are invalid.</td>
</tr>
<tr>
<td>WHLPSBUSY</td>
<td>Session is busy.</td>
</tr>
<tr>
<td>WHLSSYSEXERROR</td>
<td>Function failed due to a system error.</td>
</tr>
<tr>
<td>WHLLCANCEL</td>
<td>Asynchronous function was cancelled.</td>
</tr>
</tbody>
</table>

**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.

5.2.2.1.6 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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data String</td>
<td>NA</td>
</tr>
<tr>
<td>Data Length</td>
<td>NA</td>
</tr>
<tr>
<td>PS Position</td>
<td>NA</td>
</tr>
</tbody>
</table>

Return Codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WHLLOK</td>
<td>The PS is uninhibited and ready for input.</td>
</tr>
<tr>
<td>WHLNOTCONNECTED</td>
<td>Your WinHLLAPI application is not connected to a valid host session.</td>
</tr>
<tr>
<td>WHLLPSBUSY</td>
<td>Function timed out while still inhibited.</td>
</tr>
<tr>
<td>WHLLHIBITED</td>
<td>The PS is inhibited.</td>
</tr>
<tr>
<td>SHLLSYSERROR</td>
<td>The function failed due to a system error.</td>
</tr>
<tr>
<td>WHLLCANCEL</td>
<td>The asynchronous function was cancelled.</td>
</tr>
</tbody>
</table>

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.

WHLLINVALID

hAsyncTask is not a valid task handle.

WHLLLALREADY

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, LPWHLLAPIDATA 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:

**WHLLOK**

Indicates that the underlying network subsystem is unavailable.

**WHLLSYSNOTREADY**

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.
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

```c
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

```c
FARPROC WinHLLAPISetBlockingHook(FARPROC lpfnBlockingHook)
```

Parameters

- `lpfnBlockingHook`: Pointer to a function.
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:

```c
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

```c
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 zhlap32.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

© 2014, Cybele Software, Inc.
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
    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
        Catch ex As Exception
            MsgBox(ex.ToString)
        End Try
    End Sub

© 2014, Cybele Software, Inc.
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)

    With AxOHIOManager1.Sessions.Item(ComboBox1.Text)
        For FieldIdx = 0 To (.Screen.Fields.Count - 1)
            ListBox1.Items.Add("   Field Nº: " + 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))
        Next FieldIdx
    End With
End Sub
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
        Button9.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 = False
        Button10.Enabled = False
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
    MsgBox(ex.ToString)
    AxOHIOManager1 = Nothing
    Exit Sub
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 + "+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>
5.3.1.3 Delphi

```pascal
unit TestX;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, zOHIO_TLB, StdCtrls, StrUtils, ComCtrls,
  Mask, ExtCtrls, uTnbHIOhioConsts;

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);
  end;

```

© 2014, Cybele Software, Inc.
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;
    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);
    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);
  end;
end;
begin
    for FieldIdx := 0 to (Screen.Fields.Count - 1) do
    begin
        Memo1.Lines.Add('   Field Nº: ' + IntToStr(FieldIdx));
        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'));
    end;
    Application.ProcessMessages;
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
    except
        on E:Exception do
            ShowMessage(E.Message);
    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
    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';
Programming Reference

383

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;
© 2014, Cybele Software, Inc.


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:

![OHIO Containment Hierarchy Diagram](image)

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:

```
OHIO: Base Class
Manager | Sessions | Session | OIA | Fields | Field | Position
```

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**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>plane</td>
<td>A valid OHIO_PLANE value.</td>
</tr>
</tbody>
</table>

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**

```delphi
TbNumberOfPosition := 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**

```vbnet
Boolean = Field.Hidden
```

**Delphi Syntax**

```delphi
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**

```vbnet
Boolean = Field.HighIntensity
```

**Delphi Syntax**

```delphi
Boolean := Field.HighIntensity;
```

**Remarks**

Returns True if the field is high intensity, otherwise False.

Returns the length of the field.

**Visual Basic Syntax**

```vbnet
Integer = Field.Length
```

**Delphi Syntax**

```delphi
```
[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**

**Delphi Syntax**

**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**

**Delphi Syntax**
**Remarks**

This property returns True if the field is pen-selectable, otherwise False.

Indicates whether or not the field is protected.

**Visual Basic Syntax**

```vbnet
[Boolean =] Field.Protected
```

**Delphi Syntax**

```delphi
[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**

```vbnet
[TTnbOhioPosition =] Field.StartPos
```

**Delphi Syntax**

```delphi
[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**

```vbnet
Field.String [=} String]
```

**Delphi Syntax**

```delphi
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**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TargetPosition</td>
<td>The target row and column.</td>
</tr>
</tbody>
</table>
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**

```vbnet
[TTnbOhioField =] Fields.FindByString(targetString As String, startPos As TTnbOhioPosition, length As Integer, dir As Integer, ignoreCase As Boolean)
```

**Delphi Syntax**

```delphi
[TTnbOhioField :=] Fields.FindByString(targetString:string; startPos: TTnbOhioPosition; length:integer; dir:integer; ignoreCase:boolean);
```

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>targetString</td>
<td>The target string.</td>
</tr>
<tr>
<td>startPos</td>
<td>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).</td>
</tr>
<tr>
<td>length</td>
<td>The length from startPos to include in the search.</td>
</tr>
<tr>
<td>dir</td>
<td>An OHIO_DIRECTION value.</td>
</tr>
<tr>
<td>ignoreCase</td>
<td>Indicates whether the search is case sensitive. True means that case will be ignored. False means the search will be case sensitive.</td>
</tr>
</tbody>
</table>

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**

```vbnet
[TTnbOhioField =] Fields.Item(index As Integer)
```

**Delphi Syntax**

```delphi
[TTnbOhioField :=] Fields.Item(index:integer);
```

**Parameters**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>index</td>
<td>The index of the target OhioSession.</td>
</tr>
</tbody>
</table>

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**

```
```

**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**

```vb
[Integer =] Fields.Rows
```

**Delphi Syntax**

```delphi
[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**

```vb
Manager.CloseSession(SessionObject As TTnbOhioSession)
Manager.CloseSession(SessionName As String)
```

**Delphi Syntax**

```delphi
Manager.CloseSession(SessionObject:TTnbOhioSession);
Manager.CloseSession(SessionName:string);
```
Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SessionObject</td>
<td>The OhioSession to close.</td>
</tr>
<tr>
<td>SessionName</td>
<td>The SessionName of the OhioSession to close.</td>
</tr>
</tbody>
</table>

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**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConfigurationResource</td>
<td>A vendor specific string used to provide configuration information</td>
</tr>
<tr>
<td>SessionName</td>
<td>The unique name associated with an OhioSession</td>
</tr>
</tbody>
</table>

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**

```vbnet
[OhioSessions =] Manager.Sessions
```

**Delphi Syntax**

```delphi
```

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**

```vbnet
[Boolean=] OIA.Alphanumeric
```
**Delphi Syntax**

```delphi
[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**

```vbnet
[Integer =] OIA.CommCheckCode
```

**Delphi Syntax**

```delphi
[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**

```vbnet
[Integer =] OIA.InputInhibited
```

**Delphi Syntax**

```delphi
[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**

```vbnet
[Integer =] OIA.MachineCheckCode
```

**Delphi Syntax**
In the BGI:

\[ \text{Object} = \text{OIA}. \text{MachineCheckCode}; \]

Remarks

If InputInhibited returns OHIO_INPUTINHIBITED_MACHINEMACHINECHECK, this property will return the machine check code.

Indicates whether the field which contains the cursor is a numeric-only field.

Visual Basic Syntax

\[ \text{Boolean} = \text{OIA}. \text{Numeric} \]

Delphi Syntax

\[ \text{Boolean} := \text{OIA}. \text{Numeric}; \]

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

\[ \text{Integer} = \text{OIA}. \text{Owner} \]

Delphi Syntax

\[ \text{Integer} := \text{OIA}. \text{Owner}; \]

Remarks

The Integer returned by this property specifies one of the \text{OHIO_OWNER} values.

Returns the program check code.

Visual Basic Syntax

\[ \text{Integer} = \text{OIA}. \text{ProgCheckCode} \]

Delphi Syntax

\[ \text{Integer} := \text{OIA}. \text{ProgCheckCode}; \]

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

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TnbFields</td>
<td>The fields to refresh.</td>
</tr>
<tr>
<td>fromPos</td>
<td>The position where to start.</td>
</tr>
</tbody>
</table>

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**

```vbnet
[TTnbOhioPosition =] Screen.FindString(TargetString As String, startPos As OhioPosition, length As Integer, dir As OHIO_DIRECTION, IgnoreCase As Boolean)
```

**Delphi Syntax**

```delphi
[TTnbOhioPosition :=] Screen.FindString(TargetString:string; startPos: TTnbOhioPosition; length:integer, dir:integer; IgnoreCase:boolean);
```

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TargetString</td>
<td>The target string.</td>
</tr>
<tr>
<td>startPos</td>
<td>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).</td>
</tr>
<tr>
<td>length</td>
<td>The length from startPos to include in the search.</td>
</tr>
<tr>
<td>dir</td>
<td>An OHIO_DIRECTION value.</td>
</tr>
<tr>
<td>IgnoreCase</td>
<td>Indicates whether the search is case sensitive. True means that case will be ignored. False means the search will be case sensitive.</td>
</tr>
</tbody>
</table>

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**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>start</td>
<td>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). &quot;start&quot; must be positionally less than &quot;end&quot;.</td>
</tr>
<tr>
<td>end</td>
<td>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). &quot;end&quot; must be positionally greater than &quot;start&quot;.</td>
</tr>
<tr>
<td>plane</td>
<td>A valid OHIO_PLANE value.</td>
</tr>
</tbody>
</table>

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**
SendAid(aidKey:integer);

Parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>aidKey</td>
<td>The aid key to send to the virtual screen.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>text</td>
<td>The string of characters to be sent.</td>
</tr>
<tr>
<td>location</td>
<td>OhioPosition where the string should be written.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>text</td>
<td>String to place in the virtual screen</td>
</tr>
<tr>
<td>location</td>
<td>Position where the string should be written</td>
</tr>
</tbody>
</table>

© 2014, Cybele Software, Inc.
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**

```vbscript
[Boolean] = Screen.Wait([TimeOut] As Integer)
```

**Delphi Syntax**

```delphi
[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**

```vbscript
[Boolean =] Screen.WaitFor(Value As String, [TimeOut] As Integer, ResetWait As Boolean)
```

```vbscript
[Integer =] Screen.WaitFor(Value As String, [TimeOut] As Integer, ResetWait As Boolean)
```

**Delphi Syntax**

```delphi
[WordBool : =] Screen.WaitFor(Value:string;[TimeOut]:Integer; ResetWait:Boolean)
```

```delphi
[Integer : =] Screen.WaitFor(const Values:Array of string;[TimeOut]:Integer; ResetWait:Boolean);
```

**Parameters**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>The string of characters expected to be found.</td>
</tr>
<tr>
<td>Values</td>
<td>The array of strings where one of them is expected to be found. In Visual Basic this is a</td>
</tr>
</tbody>
</table>
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**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TimeOut</td>
<td>Optional parameter measured in milliseconds.</td>
</tr>
</tbody>
</table>

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.WaitForNewUnlock([TimeOut] As Integer)
```

**Delphi Syntax**

```
[WordBool :=] Screen.WaitForNewUnlock([TimeOut]: Integer);
```

**Parameters**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TimeOut</td>
<td>Optional parameter measured in milliseconds.</td>
</tr>
</tbody>
</table>

This method waits for a new screen unlock to arrive 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.
[Boolean =] Screen.WaitForScreen([TimeOut] As Integer)

Delphi Syntax

[WordBool :=] Screen.WaitForScreen([TimeOut]: Integer);

Parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TimeOut</td>
<td>Optional parameter measured in milliseconds.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TimeOut</td>
<td>Optional parameter measured in milliseconds.</td>
</tr>
</tbody>
</table>

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**

\[
\text{Boolean} = \text{Session}.\text{WaitForDisconnect}([\text{TimeOut}] \text{ As Integer})
\]

**Delphi Syntax**

\[
\text{WordBool} := \text{Session}.\text{WaitForDisconnect}([\text{TimeOut}]: \text{Integer});
\]

**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**

\[
\text{String} = \text{Session}.\text{ConfigurationResource}
\]

**Delphi Syntax**

\[
\text{String} := \text{Session}.\text{ConfigurationResource};
\]

**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**

\[
\text{String } = \] Sessions.SessionState

**Delphi Syntax**

\[
\text{String } := \] Session.SessionState;

Returns the SessionType for this OhioSession object.

**Visual Basic Syntax**

\[
\text{Integer } = \] Session.SessionType

**Delphi Syntax**

\[
\text{Integer } := \] Session.SessionType;

Sets the TNBXTrace 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 TNBXSync 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.

\([= \text{value}]\) or \([:= \text{value}]\) Denotes values to be assigned.
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

\[
\begin{align*}
\text{AddSession} & \text{ (ConfigurationResource As String, SessionName As String)} \\
\text{AddSession} & \text{ (ConfigurationResource As Integer, SessionName As String)} \\
\text{AddSession} & \text{ (ConfigurationResource As TTnbComApi, SessionName As String)}
\end{align*}
\]

Delphi Syntax

\[
\begin{align*}
\text{AddSession} & \text{ (ConfigurationResource:string; SessionName:string)} \\
\text{AddSession} & \text{ (ConfigurationResource:integer; SessionName:string)} \\
\text{AddSession} & \text{ (ConfigurationResource:TTnbComApi; SessionName:string)}
\end{align*}
\]

Parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConfigurationResource</td>
<td>Full path of the Configuration File (.ZCC) to use for</td>
</tr>
</tbody>
</table>
host connection information. This parameter can be a string, an integer or a TTnbComApi type.

| SessionName | 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

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SessionObject</td>
<td>The OhioSession to close.</td>
</tr>
<tr>
<td>SessionName</td>
<td>The SessionName of the OhioSession to close.</td>
</tr>
</tbody>
</table>

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**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>index</td>
<td>The index of the target OhioSession.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Aid Key Constants</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>OHIO_AID_KEY_ATTN</td>
<td>$0182</td>
</tr>
<tr>
<td>OHIO_AID_KEY_BACKSPACE</td>
<td>$0183</td>
</tr>
<tr>
<td>OHIO_AID_KEY_BACKTAB</td>
<td>$0184</td>
</tr>
<tr>
<td>OHIO_AID_KEY_CLEAR</td>
<td>$0186</td>
</tr>
<tr>
<td>OHIO_AID_KEY_CURSORSEL</td>
<td>$0188</td>
</tr>
<tr>
<td>OHIO_AID_KEY_DELETE</td>
<td>$0189</td>
</tr>
<tr>
<td>OHIO_AID_KEY_DOWN</td>
<td>$018B</td>
</tr>
<tr>
<td>OHIO_AID_KEY_DUP</td>
<td>$018C</td>
</tr>
<tr>
<td>OHIO_AID_KEY_ENTER</td>
<td>$018D</td>
</tr>
<tr>
<td>OHIO_AID_KEY_ERASEEOF</td>
<td>$018E</td>
</tr>
<tr>
<td>OHIO_AID_KEY_ERASEINPUT</td>
<td>$018F</td>
</tr>
<tr>
<td>OHIO_AID_KEY_HOME</td>
<td>$0190</td>
</tr>
<tr>
<td>OHIO_AID_KEY_LEFT</td>
<td>$0193</td>
</tr>
<tr>
<td>OHIO_AID_KEY_MARK</td>
<td>$0194</td>
</tr>
<tr>
<td>OHIO_AID_KEY_NEWLINE</td>
<td>$0195</td>
</tr>
<tr>
<td>OHIO_AID_KEY_PA1</td>
<td>$0196</td>
</tr>
<tr>
<td>OHIO_AID_KEY_PA2</td>
<td>$0197</td>
</tr>
<tr>
<td>OHIO_AID_KEY_PA3</td>
<td>$0198</td>
</tr>
<tr>
<td>Aid Key Constants</td>
<td>Value</td>
</tr>
<tr>
<td>-------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>OHIO_AID_KEY_ATTN</td>
<td>$0182</td>
</tr>
<tr>
<td>OHIO_AID_KEY_BACKSPACE</td>
<td>$0183</td>
</tr>
<tr>
<td>OHIO_AID_KEY_BACKTAB</td>
<td>$0184</td>
</tr>
<tr>
<td>OHIO_AID_KEY_CLEAR</td>
<td>$0186</td>
</tr>
<tr>
<td>OHIO_AID_KEY_CURSORSEL</td>
<td>$0188</td>
</tr>
<tr>
<td>OHIO_AID_KEY_DELETE</td>
<td>$0189</td>
</tr>
<tr>
<td>OHIO_AID_KEY_DOWN</td>
<td>$018B</td>
</tr>
<tr>
<td>OHIO_AID_KEY_DUP</td>
<td>$018C</td>
</tr>
<tr>
<td>OHIO_AID_KEY_ENTER</td>
<td>$018D</td>
</tr>
<tr>
<td>OHIO_AID_KEY_ERASEEOF</td>
<td>$018E</td>
</tr>
<tr>
<td>OHIO_AID_KEY_ERASEINPUT</td>
<td>$018F</td>
</tr>
<tr>
<td>OHIO_AID_KEY_HOME</td>
<td>$0190</td>
</tr>
<tr>
<td>OHIO_AID_KEY_LEFT</td>
<td>$0193</td>
</tr>
<tr>
<td>OHIO_AID_KEY_MARK</td>
<td>$0194</td>
</tr>
</tbody>
</table>

5.3.4.2.2 5250 Key Values
5.3.4.2.3 Ohio Constants

Constants for all Ohio classes.

**OHIO_INFO**

<table>
<thead>
<tr>
<th>Constant</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OHIO_VERSION_LEVEL</td>
<td>OHIO 1.00</td>
<td>The OHIO version level of this implementation. The form is &quot;OHIO nn.nn&quot;</td>
</tr>
<tr>
<td>OHIO_VENDOR_NAME</td>
<td>Cybele Software Inc.</td>
<td>The name of the vendor providing this OHIO implementation. Format is vendor defined.</td>
</tr>
<tr>
<td>OHIO_VENDOR_PRODUCT_VERSION</td>
<td>TN Bridge Integration Pack 3.0</td>
<td>The vendor product version that is providing the OHIO implementation. Format is vendor specific.</td>
</tr>
</tbody>
</table>

**OHIO_SESSION cfg keys**

<table>
<thead>
<tr>
<th>Constant</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OHIO_SESSION_TYPE</td>
<td>OHIO_SESSION_TYPE=</td>
<td>The OHIO Session Type: unknown host, 3270 or 5250 host.</td>
</tr>
<tr>
<td>OHIO_SESSION_HOST</td>
<td>OHIO_SESSION_HOST=</td>
<td>The OHIO Session Host.</td>
</tr>
<tr>
<td>OHIO_SESSION_PORT</td>
<td>OHIO_SESSION_PORT=</td>
<td>The OHIO Session Port.</td>
</tr>
<tr>
<td>OHIO_SESSION_NAME</td>
<td>OHIO_SESSION_NAME=</td>
<td>The OHIO Session Name.</td>
</tr>
<tr>
<td>OHIO_SESSION_CONFIG_RESOURCE</td>
<td>OHIO_SESSION_CONFIG_RESOURCE=</td>
<td>The OHIO Session Configuration Resource.</td>
</tr>
<tr>
<td>OHIO_SESSION_TN_ENHANCED</td>
<td>OHIO_SESSION_TN_ENHANCED=</td>
<td>The Screen Size for the OHIO Session.</td>
</tr>
<tr>
<td>OHIO_SESSION_SCREEN_SIZE</td>
<td>OHIO_SESSION_SCREEN_SIZE=</td>
<td>The Screen Size for the OHIO Session.</td>
</tr>
<tr>
<td>OHIO_SESSION_CODE_PAGE</td>
<td>OHIO_SESSION_CODE_PAGE=</td>
<td>The Code Page for the OHIO Session.</td>
</tr>
</tbody>
</table>
### OHIO_DIRECTION

<table>
<thead>
<tr>
<th>Constant</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OHIO_DIRECTION_FORWARD</td>
<td>0</td>
<td>Forward (beginning towards end)</td>
</tr>
<tr>
<td>OHIO_DIRECTION_BACKWARD</td>
<td>1</td>
<td>Backward (end towards beginning)</td>
</tr>
</tbody>
</table>

### OHIO_TYPE

<table>
<thead>
<tr>
<th>Constant</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OHIO_TYPE_UNKNOWN</td>
<td>0</td>
<td>Unknown Host</td>
</tr>
<tr>
<td>OHIO_TYPE_3270</td>
<td>1</td>
<td>3270 Host</td>
</tr>
<tr>
<td>OHIO_TYPE_5250</td>
<td>2</td>
<td>5250 Host</td>
</tr>
</tbody>
</table>

### OHIO_STATE

<table>
<thead>
<tr>
<th>Constant</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OHIO_STATE_DISCONNECTED</td>
<td>0</td>
<td>The communication link to the host is disconnected</td>
</tr>
<tr>
<td>OHIO_STATE_CONNECTED</td>
<td>1</td>
<td>The communication link to the host is connected</td>
</tr>
</tbody>
</table>

### OHIO_PLANE

<table>
<thead>
<tr>
<th>Constant</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OHIO_PLANE_TEXT</td>
<td>1</td>
<td>Indicates Plane Text (character data)</td>
</tr>
<tr>
<td>OHIO_PLANE_COLOR</td>
<td>2</td>
<td>Indicates Plane Color (standard HLLAPI CGA color values)</td>
</tr>
<tr>
<td>OHIO_PLANE_FIELD</td>
<td>4</td>
<td>Indicates Field Attribute Plane (field attribute bytes)</td>
</tr>
<tr>
<td>OHIO_PLANE_EXTENDED</td>
<td>8</td>
<td>Indicates Extended Plane (extended attribute bytes)</td>
</tr>
</tbody>
</table>

### OHIO_COLOR

<table>
<thead>
<tr>
<th>Constant</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OHIO_COLOR_BLACK</td>
<td>0</td>
<td>Black</td>
</tr>
<tr>
<td>OHIO_COLOR_BLUE</td>
<td>1</td>
<td>Blue</td>
</tr>
<tr>
<td>OHIO_COLOR_GREEN</td>
<td>2</td>
<td>Green</td>
</tr>
<tr>
<td>OHIO_COLOR_CYAN</td>
<td>3</td>
<td>Cyan</td>
</tr>
<tr>
<td>OHIO_COLOR_RED</td>
<td>4</td>
<td>Red</td>
</tr>
<tr>
<td>OHIO_COLOR_MAGENTA</td>
<td>5</td>
<td>Magenta</td>
</tr>
<tr>
<td>OHIO_COLOR_YELLOW</td>
<td>6</td>
<td>Yellow</td>
</tr>
<tr>
<td>OHIO_COLOR_WHITE</td>
<td>7</td>
<td>White</td>
</tr>
</tbody>
</table>

### OHIO_EXTENDED
### OHIO_EXTENDED

<table>
<thead>
<tr>
<th>Constant</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OHIO_EXTENDED_HILITE</td>
<td>$C0</td>
<td>Bitmask for Highlighting Bits</td>
</tr>
<tr>
<td>OHIO_EXTENDED_COLOR</td>
<td>$38</td>
<td>Bitmask for Color Bits</td>
</tr>
<tr>
<td>OHIO_EXTENDED_RESERVED</td>
<td>$07</td>
<td>Bitmask for Reserved Bits</td>
</tr>
<tr>
<td>OHIO_EXTENDED_HILITE_NORMAL</td>
<td>$00</td>
<td>Normal highlighting</td>
</tr>
<tr>
<td>OHIO_EXTENDED_HILITE_BLINK</td>
<td>$40</td>
<td>Blinking highlighting</td>
</tr>
<tr>
<td>OHIO_EXTENDED_HILITE_REVERSE</td>
<td>$80</td>
<td>Reverse video highlighting</td>
</tr>
<tr>
<td>OHIO_EXTENDED_HILITE_UNDERLINE</td>
<td>$C0</td>
<td>Under score highlighting</td>
</tr>
<tr>
<td>OHIO_EXTENDED_COLOR_DEFAULT</td>
<td>$00</td>
<td>Default color</td>
</tr>
<tr>
<td>OHIO_EXTENDED_COLOR_BLUE</td>
<td>$08</td>
<td>Blue</td>
</tr>
<tr>
<td>OHIO_EXTENDED_COLOR_RED</td>
<td>$10</td>
<td>Red</td>
</tr>
<tr>
<td>OHIO_EXTENDED_COLOR_PINK</td>
<td>$18</td>
<td>Pink</td>
</tr>
<tr>
<td>OHIO_EXTENDED_COLOR_GREEN</td>
<td>$20</td>
<td>Green</td>
</tr>
<tr>
<td>OHIO_EXTENDED_COLOR_TURQUOISE</td>
<td>$28</td>
<td>Turquoise</td>
</tr>
<tr>
<td>OHIO_EXTENDED_COLOR_YELLOW</td>
<td>$30</td>
<td>Yellow</td>
</tr>
<tr>
<td>OHIO_EXTENDED_COLOR_WHITE</td>
<td>$38</td>
<td>White</td>
</tr>
<tr>
<td>OHIO_EXTENDED_COLUMN_SEPARATOR</td>
<td>$02</td>
<td>Separator</td>
</tr>
</tbody>
</table>

### OHIO_FIELD

<table>
<thead>
<tr>
<th>Constant</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OHIO_FIELD_ATTRIBUTE</td>
<td>$C0</td>
<td>Bitmask for field attribute</td>
</tr>
<tr>
<td>OHIO_FIELD_PROTECTED</td>
<td>$20</td>
<td>Protected field</td>
</tr>
<tr>
<td>OHIO_FIELD_NUMERIC</td>
<td>$10</td>
<td>Numeric field</td>
</tr>
<tr>
<td>OHIO_FIELD_HIDDEN</td>
<td>$0C</td>
<td>Hidden field</td>
</tr>
<tr>
<td>OHIO_FIELD_PEN_SELECTABLE</td>
<td>$08</td>
<td>Pen selectable field</td>
</tr>
<tr>
<td>OHIO_FIELD_HIGH_INTENSITY</td>
<td>$04</td>
<td>High Intensity field</td>
</tr>
<tr>
<td>OHIO_FIELD_RESERVED</td>
<td>$02</td>
<td>Reserved field</td>
</tr>
<tr>
<td>OHIO_FIELD_MODIFIED</td>
<td>$01</td>
<td>Modified field</td>
</tr>
</tbody>
</table>

### OHIO_UPDATE

<table>
<thead>
<tr>
<th>Constant</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OHIO_UPDATE_CLIENT</td>
<td>0</td>
<td>Update initiated by client</td>
</tr>
<tr>
<td>OHIO_UPDATE_HOST</td>
<td>1</td>
<td>Update initiated by host</td>
</tr>
</tbody>
</table>

### OHIO_OWNER

<table>
<thead>
<tr>
<th>Constant</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OHIO_OWNER_UNKNOWN</td>
<td>0</td>
<td>Uninitialized</td>
</tr>
</tbody>
</table>
### OHIO_OWNER_APP
- **1**: Application or 5250 host
- **2**: 3270 - Myjob
- **3**: NVT (Network Virtual Terminal) mode
- **4**: Unowned (3270 only)
- **5**: SSCP (3270 only)

### OHIO_INPUTINHIBITED
<table>
<thead>
<tr>
<th>Constant</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OHIO_INPUTINHIBITED_NOTINHIBITED</td>
<td>0</td>
<td>Input not inhibited</td>
</tr>
<tr>
<td>OHIO_INPUTINHIBITED_SYSTEM_WAIT</td>
<td>1</td>
<td>Input inhibited by a System Wait state (&quot;X SYSTEM&quot; or &quot;X []&quot;)</td>
</tr>
<tr>
<td>OHIO_INPUTINHIBITED_COMMCHECK</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>OHIO_INPUTINHIBITED_PROGCHECK</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>OHIO_INPUTINHIBITED_MACHINECHECK</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>OHIO_INPUTINHIBITED_OTHER</td>
<td>5</td>
<td>Input inhibited by something other than above states</td>
</tr>
</tbody>
</table>

### OHIO_SESSION_STATE
<table>
<thead>
<tr>
<th>Constant</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OHIO_SESSION_STATE_NOSESSION</td>
<td>0</td>
<td>No session established</td>
</tr>
<tr>
<td>OHIO_SESSION_STATE_SSCPCLU</td>
<td>1</td>
<td>SSCPLU session state</td>
</tr>
<tr>
<td>OHIO_SESSION_STATE_LULU</td>
<td>2</td>
<td>LULU session state</td>
</tr>
</tbody>
</table>

### OHIO_EXCEPTION
<table>
<thead>
<tr>
<th>Constant</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OHIO_EXCEPTION_SESSIONALREADYEXISTS</td>
<td></td>
<td>Already exists a session with that name</td>
</tr>
<tr>
<td>OHIO_EXCEPTION_BADCFGFILE</td>
<td></td>
<td>Error in configuration file</td>
</tr>
<tr>
<td>OHIO_EXCEPTION_BADHOSTNAME</td>
<td></td>
<td>There is no data provided in '+OHIO_SESSION_HOST+' configuration field</td>
</tr>
<tr>
<td>OHIO_EXCEPTION_BADPORTNUMBER</td>
<td></td>
<td>Data provided in '+OHIO_SESSION_PORT+' configuration field is not a number</td>
</tr>
<tr>
<td>OHIO_EXCEPTION_INVALIDPORTNUMBER</td>
<td></td>
<td>Data provided in '+OHIO_SESSION_PORT+' configuration field is out of range</td>
</tr>
<tr>
<td>OHIO_EXCEPTION_TOOMANYSESSIONS</td>
<td></td>
<td>Too many sessions</td>
</tr>
</tbody>
</table>
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

\[
\text{TTnbOhioPosition = Object.CreateOhioPosition(row As Integer, col As Integer)}\]

Delphi Syntax

\[
\text{TTnbOhioPosition := Object.CreateOhioPosition(row, col:integer)};\]

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>row</td>
<td>The row coordinate</td>
</tr>
<tr>
<td>col</td>
<td>The column coordinate</td>
</tr>
</tbody>
</table>

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

\[
\text{String = Object.OhioVersion}\]
**Delphi Syntax**

```delphi
```

### 5.3.4.3.2.2 VendorName

Returns the name of the vendor providing this OHIO implementation. Format is vendor defined.

**Visual Basic Syntax**

```vbnet
[String =] OhioObject.VendorName
```

**Delphi Syntax**

```delphi
[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**

```vbnet
[TTelnet =] Screen_VENDORObject
[TTelnet =] SessionVendorObject
[TOIA =] OIA_VENDORObject
[TSessions =] Sessions_VENDORObject
[TFIELDS =] FieldVendorObject
```

**Delphi Syntax**

```delphi
[TTelnet :=] ScreenVendorObject;
[TTelnet :=] SessionVendorObject;
[TOIA :=] OIAVendorObject;
[TSessions :=] SessionsVendorObject;
[TFIELDS :=] FieldVendorObject;
```

### 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 =] $\textit{OhioObject}.\textbf{VendorProductVersion}$

**Delphi Syntax**


5.3.4.3.3 Syntax Conventions

The following text formats are used throughout the Components Reference:

*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=\textit{zScope's exe name [zClassic, zExpress3270, etc]}, Service=\textit{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 \textit{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:**

<table>
<thead>
<tr>
<th>Host Function Key</th>
<th>Mnemonic*</th>
<th>5250</th>
<th>3270</th>
</tr>
</thead>
<tbody>
<tr>
<td>@</td>
<td>@@</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Alt</td>
<td>@A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attention</td>
<td>@A@Q</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Backspace</td>
<td>@&lt;</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Backtab (Left Tab)</td>
<td>@B</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Clear</td>
<td>@C</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Cursor Down</td>
<td>@V</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Cursor Left</td>
<td>@L</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Cursor Right</td>
<td>@Z</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Cursor Select</td>
<td>@A@J</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Cursor Up</td>
<td>@U</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Delete</td>
<td>@D</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Dup</td>
<td>@S@x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>End</td>
<td>@q</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Enter</td>
<td>@E</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Erase EOF</td>
<td>@F</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Erase Input</td>
<td>@A@F</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Field Exit</td>
<td>@A@E</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Field Mark</td>
<td>@S@y</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Field -</td>
<td>@A@-</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Field +</td>
<td>@A@+</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Help</td>
<td>@H</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Home</td>
<td>@0 (zero)</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Insert</td>
<td>@I</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Insert Toggle</td>
<td>@A@I</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Host Print</td>
<td>@P</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Left Tab (Backtab)</td>
<td>@B</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>New Line</td>
<td>@N</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Page Up</td>
<td>@u</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Page Down</td>
<td>@v</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Record Backspace</td>
<td>@A@&lt;</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Right Tab (Tab)</td>
<td>@T</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Shift</td>
<td>@S</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sys Request</td>
<td>@A@H</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Tab (Right Tab)</td>
<td>@T</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Test</td>
<td>@A@C</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>PA1</td>
<td>@x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PA2</td>
<td>@y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PA3</td>
<td>@z</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Key</td>
<td>Symbol</td>
<td>Value</td>
<td>Value</td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>PF1/F1</td>
<td>@1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PF2/F2</td>
<td>@2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PF3/F3</td>
<td>@3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PF4/F4</td>
<td>@4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PF5/F5</td>
<td>@5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PF6/F6</td>
<td>@6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PF7/F7</td>
<td>@7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PF8/F8</td>
<td>@8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PF9/F9</td>
<td>@9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PF10/F10</td>
<td>@a</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PF11/F11</td>
<td>@b</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PF12/F12</td>
<td>@c</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PF13</td>
<td>@d</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PF14</td>
<td>@e</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PF15</td>
<td>@f</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PF16</td>
<td>@g</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PF17</td>
<td>@h</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PF18</td>
<td>@i</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PF19</td>
<td>@j</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PF20</td>
<td>@k</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PF21</td>
<td>@l</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PF22</td>
<td>@m</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PF23</td>
<td>@n</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PF24</td>
<td>@o</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* 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:

<table>
<thead>
<tr>
<th>Country</th>
<th>Code Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Albania</td>
<td>CP500</td>
</tr>
<tr>
<td>Arabic Countries</td>
<td>CP420</td>
</tr>
<tr>
<td>Australia</td>
<td>CP037</td>
</tr>
<tr>
<td>Austria Euro</td>
<td>CP273</td>
</tr>
<tr>
<td>Austria</td>
<td>CP1141</td>
</tr>
<tr>
<td>Belarus Euro</td>
<td>CP1154</td>
</tr>
<tr>
<td>Belarus</td>
<td>CP1025</td>
</tr>
<tr>
<td>Belarus/EEUU</td>
<td>CP037</td>
</tr>
<tr>
<td>Belgium Euro</td>
<td>CP1148</td>
</tr>
<tr>
<td>Belgium</td>
<td>CP500</td>
</tr>
<tr>
<td>Belgium</td>
<td>CP037</td>
</tr>
<tr>
<td>Bosnia/Herzegovina</td>
<td>CP870</td>
</tr>
<tr>
<td>Brazil Euro</td>
<td>CP1140</td>
</tr>
<tr>
<td>Brazil</td>
<td>CP037</td>
</tr>
<tr>
<td>Bulgaria Euro</td>
<td>CP1154</td>
</tr>
<tr>
<td>Bulgaria</td>
<td>CP1125</td>
</tr>
<tr>
<td>Canada Euro</td>
<td>CP1140</td>
</tr>
<tr>
<td>Canada</td>
<td>CP037</td>
</tr>
<tr>
<td>Canada</td>
<td>CP1047</td>
</tr>
<tr>
<td>Croatia Euro</td>
<td>CP1153</td>
</tr>
<tr>
<td>Croatia</td>
<td>CP870</td>
</tr>
<tr>
<td>Czech Euro</td>
<td>CP1153</td>
</tr>
<tr>
<td>Czech</td>
<td>CP870</td>
</tr>
<tr>
<td>Denmark Euro</td>
<td>CP1142</td>
</tr>
<tr>
<td>Denmark</td>
<td>CP227</td>
</tr>
<tr>
<td>Estonia Euro</td>
<td>CP1157</td>
</tr>
<tr>
<td>Estonia</td>
<td>CP1122</td>
</tr>
<tr>
<td>Estonia</td>
<td>CP037</td>
</tr>
<tr>
<td>Finland Euro</td>
<td>CP1143</td>
</tr>
<tr>
<td>Finland</td>
<td>CP278</td>
</tr>
<tr>
<td>France Euro</td>
<td>CP1143</td>
</tr>
<tr>
<td>France</td>
<td>CP297</td>
</tr>
<tr>
<td>Germany Euro</td>
<td>CP1141</td>
</tr>
<tr>
<td>Germany</td>
<td>CP273</td>
</tr>
<tr>
<td>Greece Euro</td>
<td>CP874</td>
</tr>
<tr>
<td>Greece</td>
<td>CP423</td>
</tr>
<tr>
<td>Greece</td>
<td>CP875</td>
</tr>
<tr>
<td>Hebrew</td>
<td>CP424</td>
</tr>
<tr>
<td>Hungary euro</td>
<td>CP1153</td>
</tr>
<tr>
<td>Hungary</td>
<td>CP870</td>
</tr>
<tr>
<td>International</td>
<td>CP850</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------</td>
</tr>
<tr>
<td>Iceland Euro</td>
<td>CP1149</td>
</tr>
<tr>
<td>Iceland</td>
<td>CP871</td>
</tr>
<tr>
<td>Italy Euro</td>
<td>CP1144</td>
</tr>
<tr>
<td>Italy</td>
<td>CP280</td>
</tr>
<tr>
<td>Latin America Euro</td>
<td>CP1145</td>
</tr>
<tr>
<td>Latin America</td>
<td>CP284</td>
</tr>
<tr>
<td>Latvia Euro</td>
<td>CP1156</td>
</tr>
<tr>
<td>Latvia</td>
<td>CP1112</td>
</tr>
<tr>
<td>Latvia</td>
<td>CP037</td>
</tr>
<tr>
<td>Lithuania Euro</td>
<td>CP1156</td>
</tr>
<tr>
<td>Lithuania</td>
<td>CP1112</td>
</tr>
<tr>
<td>Lithuania</td>
<td>CP037</td>
</tr>
<tr>
<td>Macedonia Euro</td>
<td>CP1154</td>
</tr>
<tr>
<td>Macedonia</td>
<td>CP1025</td>
</tr>
<tr>
<td>Multinational Euro</td>
<td>CP1148</td>
</tr>
<tr>
<td>Multinational</td>
<td>CP924</td>
</tr>
<tr>
<td>Multinational ISO</td>
<td>CP500</td>
</tr>
<tr>
<td>Netherlands Euro</td>
<td>CP1140</td>
</tr>
<tr>
<td>Netherlands</td>
<td>CP037</td>
</tr>
<tr>
<td>Norway Euro</td>
<td>CP1142</td>
</tr>
<tr>
<td>Norway</td>
<td>CP277</td>
</tr>
<tr>
<td>Poland Euro</td>
<td>CP1153</td>
</tr>
<tr>
<td>Poland</td>
<td>CP870</td>
</tr>
<tr>
<td>Portugal Euro</td>
<td>CP1140</td>
</tr>
<tr>
<td>Portugal</td>
<td>CP037</td>
</tr>
<tr>
<td>Romania Euro</td>
<td>CP1153</td>
</tr>
<tr>
<td>Romania</td>
<td>CP870</td>
</tr>
<tr>
<td>Russian Euro</td>
<td>CP1154</td>
</tr>
<tr>
<td>Russian</td>
<td>CP1025</td>
</tr>
<tr>
<td>Serbia Cyrillic Euro</td>
<td>CP1154</td>
</tr>
<tr>
<td>Serbia Latin Euro</td>
<td>CP1153</td>
</tr>
<tr>
<td>Serbia/Montenegro</td>
<td>CP1025</td>
</tr>
<tr>
<td>Slovenia Euro</td>
<td>CP1153</td>
</tr>
<tr>
<td>Slovenia</td>
<td>CP870</td>
</tr>
<tr>
<td>Slovakia Euro</td>
<td>CP1153</td>
</tr>
<tr>
<td>Slovakia</td>
<td>CP870</td>
</tr>
<tr>
<td>Spain Euro</td>
<td>CP1145</td>
</tr>
<tr>
<td>Spain</td>
<td>CP284</td>
</tr>
<tr>
<td>Sweden Euro</td>
<td>CP1143</td>
</tr>
<tr>
<td>Sweden</td>
<td>CP287</td>
</tr>
<tr>
<td>Switzerland Euro</td>
<td>CP1148</td>
</tr>
<tr>
<td>Switzerland</td>
<td>CP500</td>
</tr>
<tr>
<td>Turkey Euro</td>
<td>CP1155</td>
</tr>
<tr>
<td>Turkey</td>
<td>CP1026</td>
</tr>
<tr>
<td>Ukraine Euro</td>
<td>CP1158</td>
</tr>
<tr>
<td>Ukraine</td>
<td>CP1123</td>
</tr>
</tbody>
</table>
6.2 Modifying the Character Table

To modify the EBCDIC to ASCII conversion for a particular character, select and right-click the character you want to change and then select the 'Change Character' option from the context pop-up menu.

The Character Table dialog will be displayed, allowing you to select a new character by double-clicking on it.
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:

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>^</td>
<td>Matches the position at the beginning of the string.</td>
<td>^B matches &quot;B&quot; but only if it is the first character in the string.</td>
</tr>
<tr>
<td>$</td>
<td>Matches the position at the end of the string.</td>
<td>$p matches &quot;p&quot; but only if it is the last character in the string.</td>
</tr>
<tr>
<td>.</td>
<td>Matches any single character.</td>
<td>le. matches &quot;leg&quot; and &quot;let&quot;.</td>
</tr>
<tr>
<td>+</td>
<td>Matches the preceding character 1 or more times.</td>
<td>ca+t matches &quot;cat&quot; and &quot;caat&quot; but not &quot;ct&quot;.</td>
</tr>
<tr>
<td>*</td>
<td>Matches the preceding character 0 or more times.</td>
<td>ca*t matches &quot;ct&quot;, &quot;cat&quot;, &quot;caat&quot; and so on.</td>
</tr>
<tr>
<td>?</td>
<td>Matches the preceding character 0 or 1 times.</td>
<td>si?t matches &quot;st&quot; and &quot;sit&quot; only.</td>
</tr>
<tr>
<td>[xyz]</td>
<td>Matches any one of the enclosed characters (character set).</td>
<td>[gdp]ot matches the &quot;got&quot;, &quot;dot&quot; and &quot;pot&quot;.</td>
</tr>
<tr>
<td>[^xyz]</td>
<td>Matches any character not enclosed (complementary set).</td>
<td>[^aeiou] matches any character that is not a vocal.</td>
</tr>
<tr>
<td>[x</td>
<td>y]</td>
<td>Matches either x or y.</td>
</tr>
<tr>
<td>[a-z]</td>
<td>Matches any character in the specified range (character range).</td>
<td>[a-z] matches any lowercase letter of the alphabet.</td>
</tr>
<tr>
<td>[^a-z]</td>
<td>Matches any character not in the specified range (complementary range).</td>
<td>[^a-z] matches any character that is not in the alphabet.</td>
</tr>
<tr>
<td>\b</td>
<td>Matches a word boundary (the position between a word and a space).</td>
<td>a\l\b matches the &quot;al&quot; in &quot;general&quot; but not the &quot;al&quot; in &quot;fall&quot;.</td>
</tr>
<tr>
<td>\B</td>
<td>Matches a nonword boundary.</td>
<td>a\l\B matches the &quot;al&quot; in &quot;fall&quot; but not the &quot;al&quot; in &quot;general&quot;.</td>
</tr>
<tr>
<td>\s</td>
<td>Matches any white space character including space, tab, form-feed, and so on.</td>
<td></td>
</tr>
<tr>
<td>\S</td>
<td>Matches any non-white space character.</td>
<td></td>
</tr>
</tbody>
</table>
Appendix B - Regular Expressions

<table>
<thead>
<tr>
<th>\d</th>
<th>Matches a digit character. Equivalent to [0-9].</th>
</tr>
</thead>
<tbody>
<tr>
<td>\D</td>
<td>Matches any non-digit character. Equivalent to [^0-9].</td>
</tr>
<tr>
<td>\w</td>
<td>Matches any word character including underscore. Equivalent to [A-Za-z0-9_].</td>
</tr>
<tr>
<td>\W</td>
<td>Matches any non-word character. Equivalent to [^A-Za-z0-9_].</td>
</tr>
<tr>
<td>{n}</td>
<td>Matches a character exactly n times.</td>
</tr>
<tr>
<td>{n,}</td>
<td>Matches a character at least n times.</td>
</tr>
<tr>
<td>{n,m}</td>
<td>Matches a character at least n and at most m times.</td>
</tr>
</tbody>
</table>

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 the 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"

# 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.
**Appendix D - Web Server Configuration**

**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**

© 2014, Cybele Software, Inc.
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.


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.

![Enter Key dialog](image)

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/](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/](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](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:
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.