

ESKER *Tun*[®] *Plus*

Application Access –
32 bit Executables

Tun Plus 2014 - Version 15.0.0 Issued December 2013
Copyright © 1989-2014 Esker S.A. All rights reserved.

Copyright © 1998-2008 The OpenSSL Project. All rights reserved.
Copyright © 1995-1998 Eric Young (eay@cryptsoft.com). All rights reserved.
Copyright © 1995-2005 The Cryptix Foundation Limited. All rights reserved.
Copyright © 1995 Tatu Ylonen <ylo@cs.hut.fi>, Espoo, Finland. All rights reserved
Copyright © 1998 CORE SDI S.A., Buenos Aires, Argentina. All rights reserved
Copyright © 1995, 1996 by David Mazieres <dm@lcs.mit.edu>
Copyright © 1983, 1990, 1992, 1993, 1995 The Regents of the University of California. All rights reserved.
Copyright © 1998-2003 by Neil Hodgson neilh@scintilla.org. All Rights Reserved.
For additional information, conditions of use, and disclaimers, see copyright.pdf file.
Use and duplicate only in accordance with the Software License Agreement: Tun Products.

Esker, the Esker logo, Esker Pro, Extending the Reach of Information, Tun, and Tun Emul are trademarks, registered trademarks or service marks of Esker S.A. in the U.S., France and other countries. The following are trademarks of their respective owners in the United States and other countries: Microsoft, Windows, BackOffice, MS-DOS, XENIX are registered trademarks of Microsoft Corp. Netscape and Netscape Navigator are registered trademarks of Netscape Communications Corp. IBM, AS/400, and AIX are registered trademarks of IBM Corp. SCO is a registered trademark of Caldera International, Inc. NetWare is a registered trademark of Novell, Inc. Sun, Sun Microsystems and Java are trademarks of Sun Microsystems, Inc. Oracle is a registered trademark of Oracle Corp. Informix is a registered trademark of Informix Software Inc. Sybase is a registered trademark of Sybase, Inc. Progress is a registered trademark of Progress Software Corp. All other trademarks mentioned are the property of their respective owners.

Information in this document is subject to change without notice.



[See the list of Esker locations in the world.](#)

No part of this document may be reproduced or transmitted in any form or by any means without the prior written consent of Esker S.A.

Table of Contents

| | |
|---|-----------|
| Preface | 10 |
| Introduction to Terminal Emulation | 12 |
| Operation..... | 12 |
| Transmission mode..... | 12 |
| Esker emulators..... | 12 |
| Esker's terminal emulators and NIS..... | 13 |
| Asynchronous Emulation | 14 |
| Working principle..... | 14 |
| Terminals..... | 14 |
| Settings..... | 15 |
| Settings files..... | 15 |
| Starting the asynchronous emulator..... | 16 |
| User interface..... | 16 |
| Menus..... | 16 |
| Displaying the menus..... | 17 |
| Toolbar..... | 17 |
| Status bar..... | 18 |
| Emulation over TCP/IP..... | 18 |
| Settings..... | 18 |
| Starting an emulation session..... | 19 |
| Emulation over serial link..... | 19 |
| Settings..... | 19 |
| Starting emulation..... | 20 |
| Emulation over TAPI Modem..... | 21 |
| Emulation over SSH..... | 22 |
| Settings..... | 22 |
| Sessions and configurations..... | 25 |
| Managing multiple sessions..... | 25 |
| Configurations..... | 25 |
| Saving a configuration in an archive file..... | 25 |
| Opening a configuration..... | 26 |
| Modifying the contents of a .cfz archive..... | 26 |
| Resources editor..... | 26 |
| Editing and saving .cfg configurations..... | 27 |
| Terminal display and revamping..... | 27 |
| Loading /Saving .ctx files..... | 28 |
| Emulator options..... | 28 |
| Startup and disconnection options..... | 28 |
| National keyboards..... | 29 |

| | |
|---|-----------|
| Customizing the asynchronous emulator..... | 29 |
| HP emulation characteristics..... | 29 |
| Dynamic Data Exchange..... | 30 |
| Tun EMUL and DDE..... | 31 |
| Use..... | 32 |
| Example..... | 35 |
| IBM Synchronous Emulation..... | 39 |
| Connection protocols..... | 39 |
| TN3270E Emulation..... | 40 |
| TN5250E Emulation..... | 40 |
| 3270 gateways..... | 40 |
| UNIX SNA-TCP/IP gateways..... | 41 |
| Novell Netware for SAA gateways..... | 41 |
| Microsoft SNA Server gateways..... | 42 |
| 5250 gateways..... | 43 |
| UNIX SNA-TCP/IP gateways..... | 43 |
| NetWare for SAA gateways..... | 43 |
| Microsoft SNA server gateway..... | 44 |
| Starting the synchronous emulator..... | 44 |
| User interface..... | 44 |
| Menus..... | 44 |
| Menu display..... | 45 |
| Toolbar..... | 45 |
| Status bar / OIA bar..... | 46 |
| Connection in IBM synchronous emulation..... | 47 |
| Customizing a synchronous emulation session..... | 47 |
| Options common to the Session tab..... | 48 |
| Session tab in TN3270 or TN3270E connection (3270 emulation)..... | 48 |
| Session tab in SNA Server or Netware for SAA connection (3270 emulation)..... | 49 |
| TN5250 or TN5250E Session connecting tab (emulation 5250)..... | 49 |
| Session tab (Netware for SAA or SNA Server connection on 5250 emulation)..... | 51 |
| Multiple connection Session tab..... | 52 |
| Starting sessions..... | 53 |
| The OIA bar..... | 53 |
| Content of OIA bar..... | 53 |
| Emulation session indicators (Columns 1 to 3)..... | 53 |
| Keyboard locking indicators (Columns 9 to 17)..... | 54 |
| Message indicator (5250 only) (Columns 26 -27)..... | 55 |
| APL mode indicator (Columns 32 to 41)..... | 55 |
| Shift mode indicator (Columns 42 to 44)..... | 55 |
| Insert mode indicator (Column 52)..... | 55 |
| Cursor position indicator (Columns 75 to 80)..... | 55 |

| | |
|--|-----------|
| Program Check Messages..... | 55 |
| APL Mode(3270 emulation)..... | 57 |
| Switching to the APL Mode..... | 57 |
| HLLAPI (3270 emulation only)..... | 58 |
| Using HLLAPI..... | 59 |
| Accessing data on an IBM MainFrame server from a Windows application (HLLAPI)..... | 60 |
| Customizing the emulator..... | 64 |
| Asynchronous emulator..... | 64 |
| Screen size..... | 64 |
| Scroll type..... | 64 |
| Scroll bars..... | 64 |
| Cursor coupling..... | 64 |
| Terminal centering..... | 65 |
| File..... | 65 |
| Alignment..... | 65 |
| Tile..... | 65 |
| Scroll with text..... | 65 |
| Fit to display zone..... | 65 |
| Display zone..... | 65 |
| Synchronous emulator..... | 66 |
| Image..... | 66 |
| Display frame..... | 66 |
| Choice of font..... | 66 |
| Asynchronous emulation..... | 66 |
| Synchronous emulation..... | 67 |
| Customizing colors..... | 67 |
| Asynchronous emulation..... | 67 |
| Synchronous emulation..... | 69 |
| Working with a personal function-key panel..... | 70 |
| Asynchronous emulation..... | 70 |
| Synchronous emulation..... | 71 |
| Mouse configuration (asynchronous emulation)..... | 72 |
| Adding/deleting an event..... | 72 |
| Adding/deleting/moving an instruction..... | 72 |
| Sending a string..... | 72 |
| Running a local script..... | 73 |
| Running a macro..... | 73 |
| Function keys..... | 73 |
| Other functions..... | 73 |
| Printing..... | 74 |
| Screen print..... | 74 |
| Configuration of printouts from the server..... | 74 |

| | |
|---|------------|
| Printing with template (3270/5250 emulations)..... | 75 |
| Transparent printing (asynchronous emulation)..... | 77 |
| Cut & Paste..... | 77 |
| Copy options (asynchronous emulation)..... | 78 |
| File transfer..... | 78 |
| Asynchronous emulation..... | 79 |
| 3270 synchronous emulation..... | 80 |
| Multiple file transfer (synchronous emulation)..... | 82 |
| Display of controls after recognition of a string..... | 83 |
| Terminal customization..... | 84 |
| Asynchronous emulation..... | 84 |
| Synchronous Emulation..... | 84 |
| Modifying the character table (IBM synchronous emulations)..... | 86 |
| Customizing the keyboard..... | 86 |
| Asynchronous emulation..... | 86 |
| Synchronous emulation..... | 94 |
| Save / Load keyboard..... | 96 |
| Macro automation..... | 96 |
| Recording macros..... | 96 |
| Password encryption in macros..... | 96 |
| Editing macros..... | 97 |
| Running a macro..... | 98 |
| Association of a startup or end macro..... | 98 |
| Emulation customizer..... | 98 |
| Defining a profile..... | 98 |
| Saving a customization profile..... | 99 |
| Firewall..... | 100 |
| IBM Printers Emulation..... | 101 |
| Preliminary notion: Logical Unit (LU)..... | 101 |
| 3287 Printer Emulation..... | 102 |
| 3812 Printer Emulation..... | 102 |
| Using IBM printers emulation..... | 102 |
| From the 3270 or 5250 terminal emulation application..... | 102 |
| From the 3287 or 3812 printer emulation application itself..... | 103 |
| Connection in IBM Printers Emulation..... | 104 |
| Customizing a printing session..... | 104 |
| Options common to the Session tab..... | 104 |
| Session tab in TN3270E connection (3287 emulation)..... | 104 |
| Session tab in SNA Server or Netware for SAA connection (3287 emulation)..... | 105 |
| Session tab in TN5250E connection (3812 emulation)..... | 105 |
| Session tab (Netware for SAA or SNA Server connection on 3812 emulation)..... | 107 |
| 3287 or 3812 Print Configuration..... | 108 |

| | |
|---|------------|
| Print options for the server..... | 108 |
| PC Print Configuration..... | 108 |
| 3287 or 3812 Emulation Status..... | 109 |
| Print session status..... | 110 |
| Print Commands..... | 110 |
| Log file..... | 111 |
| Hotspots..... | 112 |
| Recognition and display rules..... | 112 |
| Types of controls..... | 112 |
| Creating hotspots..... | 113 |
| List of controls per hotspot..... | 113 |
| Modifying and deleting a hotspot or control..... | 118 |
| Action editing tools on the controls..... | 119 |
| Function-Key Panel Editor..... | 123 |
| Starting the function-key panel editor..... | 123 |
| Creating a button..... | 123 |
| Additional buttons..... | 126 |
| Default Button Settings..... | 126 |
| Panel Settings and Positioning..... | 126 |
| Set Tab Order..... | 127 |
| Opening an existing function-key panel..... | 128 |
| Saving a function-key panel..... | 128 |
| Testing a key panel..... | 128 |
| Supplementary functions of the panel editor..... | 128 |
| Context menus and toolbar..... | 130 |
| Panel Settings..... | 130 |
| Button Settings..... | 131 |
| Toolbar..... | 131 |
| Emulator Automation..... | 133 |
| Macro principles..... | 133 |
| Syntax..... | 133 |
| Sample macro..... | 134 |
| Language syntax..... | 135 |
| Advanced Use of Asynchronous Emulator..... | 137 |
| The concept of emulation..... | 137 |
| Escape sequences..... | 138 |
| Content of an escape sequences file..... | 139 |
| Terminal Initialization..... | 140 |
| Sequence Headers..... | 140 |

| | |
|--|------------|
| Defining escape sequences..... | 140 |
| Examples..... | 144 |
| The function keys..... | 145 |
| Content of a function keys file..... | 145 |
| Integration of function keys in the emulator..... | 145 |
| Terminal configuration..... | 146 |
| Content of a terminal configuration file..... | 146 |
| Details..... | 147 |
| The national keyboards..... | 148 |
| Reading a .nat file..... | 148 |
| Control codes..... | 149 |
| Code conversion..... | 150 |
| Character tables..... | 151 |
| Internal character table management..... | 152 |
| Alternate character font..... | 153 |
| Using Esker actions..... | 155 |
| Quitting Esker Viewer upon server request..... | 155 |
| File transfer requested by server..... | 155 |
| Windows to UNIX copy..... | 156 |
| UNIX to Windows copy..... | 156 |
| PC programs started by server..... | 157 |
| Macro execution requested by server..... | 157 |
| Transparent printing..... | 157 |
| Actions proposed by Esker..... | 158 |
| Example 1: transparent printing on the PC's default printer via the Windows print manager..... | 158 |
| Example 2: Direct transparent printing on a printer port..... | 159 |
| Dynamically changing terminal type..... | 159 |
| Changing sessions automatically..... | 160 |
| Mouse support in UNIX applications..... | 160 |
| Principle..... | 161 |
| Provided actions..... | 161 |
| Implementation..... | 165 |
| Miscellaneous Solutions..... | 166 |
| Color attributes in emulation..... | 166 |
| 132-column emulation under Windows..... | 168 |
| Setting number of columns in a ".ctx" file..... | 168 |
| Assigning 132 columns in the ".seq" file..... | 168 |
| Emulation with 25 lines..... | 168 |
| Scancode emulation..... | 169 |
| Using scancode mode..... | 169 |

| | |
|--|------------|
| Using COM3 and COM4..... | 169 |
| Defining modem commands..... | 170 |
| Program syntax..... | 171 |
| EMULWIN..... | 171 |
| Syntax..... | 171 |
| Example..... | 172 |
| 3270_32..... | 172 |
| Syntax..... | 172 |
| Description..... | 172 |
| 3287_32..... | 173 |
| Syntax..... | 173 |
| Description..... | 173 |
| 5250_32..... | 174 |
| Syntax..... | 174 |
| Description..... | 174 |
| 3812_32..... | 174 |
| Syntax..... | 174 |
| Description..... | 174 |
| PANED32..... | 175 |
| Syntax..... | 175 |
| Description..... | 175 |
| Serial Cabling..... | 176 |
| Overview of serial ports and cables..... | 176 |
| DTE/DCE..... | 176 |
| Minimum Cabling..... | 177 |
| 9-to-25 pin conversion..... | 177 |
| Cable possibilities..... | 178 |
| Macro Examples..... | 180 |
| Index..... | 185 |

Preface

Tun EMUL - Application Access is a terminal emulation application suite: UNIX asynchronous emulation, IBM 3270 and 5250 synchronous emulation, IBM 3287 and 3812 print servers, and function-key panel editor.

The executable versions of Tun EMUL for Windows 3.x and the ActiveX version of Tun EMUL are described in separate manuals:

- Tun EMUL for Windows 3.x – Application Access
- Tun EMUL ActiveX version – Application Access

Tun EMUL is part of the Tun software product range, as shown below:

| | Components in Windows | Components in a multi-user environment |
|--|--|--|
| Esker TCP/IP Stack | TCP/IP communication stack for Windows 3.x (DLL) | N/A |
| Network Resource Access (Tun NET) | TCP/IP applications (NIS, PING, FTP Client and Server, TELNET, RSH Client and Server, TAR, WALL, TFTP, TIME) | TCP/IP applications (PING, FTP Client and Server, TELNET VT320, RSH Client, TAR, WALL) |
| Application Access (Tun EMUL) | Terminal emulator (asynchronous, IBM3270 and IBM5250 emulation, 3287/3812 printer) | Terminal emulator (asynchronous, IBM3270 and IBM5250 emulation, 3287/3812 printer) |
| TCP/IP Network Services | NIS Browser | |

Some functionalities and procedures in this manual apply only to certain platforms. In this case, the paragraph or section in question is indicated as follows:



Windows 98 only



Multi-user environment



Excluding multi-user environment

Tun EMUL for Windows is also delivered with Tun PLUS which includes all the modules mentioned above. The Tun PLUS installation procedure proposes to install Tun EMUL.

You can also install the standard or Multi-User Windows version of Tun EMUL independently of Tun PLUS.

Introduction to Terminal Emulation

A terminal is designed to interpret the data streams from a server and enable a user to communicate with this server. There are large numbers of terminals available on the market able to connect to UNIX, HP, VT servers, IBM MainFrames, mini-computers, etc.

Terminal emulation involves reproducing the behavior of a terminal on a PC. The major advantages of terminal emulation are:

- Allowing connection to a remote server from any PC containing emulation software.
- Using other applications available on the PC simultaneously with connection.
- Taking advantage of the graphic features offered by a PC Windows type environment.

Operation

The data stream from the server comprises characters and escape sequences. The characters appear as-is at the position of the cursor, while the escape sequences are generally preceded by a special character (escape) and inform the terminal that it is required to perform certain special operations such as move cursor, clear screen, change display attribute, color, etc.

The role of the terminal emulator is to correctly interpret the escape sequences from the server, in order to restore the display corresponding to that of the original terminal.

In the other communication direction (to the server), the keys struck by the user are converted into simple characters or character strings (for example, for the function keys), which are then transmitted to the server. This conversion is once again performed by the terminal emulator.

Transmission mode

Terminal emulation is possible in two data transmission modes between the PC and the server:

- Asynchronous emulation: the PC and the server exchange data in character mode, that is character by character, with no precise mechanism or exchange supervisor (hence the notion of asynchronous). This is referred to as character mode.
- Synchronous emulation: unlike asynchronous emulation, it is not destructured streams of characters that pass between the PC and the server, but rather packets of data sent alternately in one direction then the other (hence the notion of synchronous). This is referred to as screen mode.

Esker emulators

Esker has a wide range of emulations:

- Asynchronous terminal emulations, for connecting UNIX, HP, DEC, etc. servers (Telnet protocol).
- Synchronous terminal emulations, for connection to IBM MainFrame and AS/400 servers (Telnet 3270 and Telnet 5250 protocol). The Esker emulators also offer the option of connecting to IBM servers via SNA and Netware for SAA gateways.
- Printer emulations, using the PC as a print server for printers such as IBM 3287 and 3812.

This complete range of emulations enables the user to access the various types of business servers from his or her PC and improve the ergonomics of traditional applications through the powerful configuration and customization features proposed.

Esker's terminal emulators and NIS



Tun EMUL lets you access the NIS server by calling the centralized resource management application Tun NIS. Tun EMUL Setup lets you install Tun NIS.

Tun EMUL lets you access the NIS server by calling the centralized resource management application Tun NIS. Tun EMUL's Setup proposes the installation of Tun NIS.

This functionality lets you view the resources available on the network and select them directly with a simple click of the mouse. In the context of the terminal emulation provided by Esker, Tun NIS lets you conveniently access the different servers on the network and select the one to which you want to connect. The network administrator, of course, must first configure the NIS server and define the Servers resource table using the NIS Browser.

Tun NIS also lets you access objects such as images (for the screen background, for example) and function-key panels that are shared on the network.

For a full description of the NIS Browser, see the relevant section in the manuals TCP/IP Network Services if don't have the Network Resource Access module, and Tun NET - Network Resource Access, chapter *The NIS browser*.

Asynchronous Emulation

Asynchronous emulation involves the bi-directional exchange of characters between the PC and the server. There's no precise mechanism governing the exchange. You can type characters at the same time that the server sends them. The basic unit of asynchronous emulation is the character.

Esker's asynchronous emulator runs over:

- TCP/IP LAN (Winsock norm, Telnet connection protocol).
- RS232 serial link, via one of the PC's serial ports (COM1 to COM32).
- Commuted phone line, via modem.
- SSH

Note:

The emulator's 16-bit version (for PCs with Windows 3.x) provides other access types, detailed in the 16-bit version User Guide.

Working principle

Every session is associated with a terminal; this refers to the type of emulation used between the PC and the server. Esker proposes more than 20 types of asynchronous emulation which covers most of the terminals on the market. By terminal, we mean the combination of advanced parameters like escape sequences, function-keys, control codes, sent codes, etc.

Moreover, you can add customized settings to your regular session, such as screen dimensions, font used, color and characters attributes, keyboard definition, screen background, key panel editor, mouse,... This is known as revamping or redefinition of the terminal emulation.

Note:

For compatibility reasons with previous Tun versions, the context notion, which gathers all of the terminal emulation redefinition settings (colors, screen background, dimensions, fonts, hotspots...) is retained. If you have existing context files (.ctx extension files), you can keep using them by loading them when configuring the session.

Terminals

Esker provides the emulation of more than 20 terminals, covering most of the terminals available on the market. Selecting a terminal consists in choosing advanced parameters

such as escape sequences, function keys, control codes,..., that are essential to a smooth PC/server communication.

The list of available terminals is as follows:

- ANSI, ANSIX: SCO UNIX, SCO XENIX consoles
- ANSI_EUC, ANSI_JAP: SCO UNIX, SCO XENIX consoles (Japan)
- AT386: Interactive UNIX (386) console, UNIVEL
- ATO300: ALCATEL APX
- FT: Fortune
- DASHER: Data General 412
- HPTERM: Hewlett Packard (2392A)
- IBM3151, HFT: IBM RS6000
- IMP: Printer (soft copy)
- MINITEL, ISO 6429: Minitel
- TM266: Philips P90X0
- TO300, TO300850: UNISYS U6000
- TWSSDP, TWSVIP: BULL DKU 7102/7202
- VT52, VT100, VT220, VT220VT, VT320, VT520: DEC
- WYSE50, WYSE60: WYSE
- 97801, BA80_08: Siemens/Nixdorf

Settings

You can use the terminal emulator at different levels:

- To establish a simple connection, using the terminal's basic settings;
- To add colors, a screen background, to edit screen settings, to display a key panel;
- To use macros, customize the keyboard or the mouse, design hotspots;
- To adapt settings files to specific terminal emulations (adding escape sequences, national specificities,...)

Settings files

Session parameters are stored in settings files with the following file extensions:

- .ter: Contains the names of advanced settings files.
- .tab, .fun, .nat, .seq, .cod, .snd and .ses: Advanced settings files.
- .ctx: Contains display settings.

- .pan: Function-key panel settings.
- .hot: Hotspots settings.
- .cfg: Information needed to open the sessions defined in the configuration file. The settings files referenced in the .cfg file should all be located in the same directory.
- .cfz: Archive containing settings files. Unlike a .cfg file, the archive actually contains the files themselves and not just references to the files.

Starting the asynchronous emulator

Once you have started the emulator, you can:

- Open a new emulation session by selecting the type of connection (TCP/IP, RS232, modem, etc.). See the corresponding sections for the different connection types.
- Open an session (one or more preconfigured sessions). See “Sessions and configurations” on page 25.

User interface

Menus

Note:

Not all the menus and options described below are available. The menus and options described below are not all available if no session has been opened or if you're using a customized version of the emulator from which menus and/or options have been removed.

- File: Open, close, save session or configuration, create shortcut, print and print settings, quit the emulator
- Edit: Copy/paste options
- Settings: configuration, display settings, session, function-key panel, hotspot, keyboard and terminal settings, association of a startup or end macro, emulation customize, emulator options, terminal edition, reset terminal, display (full screen, function-key panel, toolbar, status bar)
- Connection: reset connection, start and stop a modem connection, enter a phone number, send predefined commands over the connection without entering them at the keyboard
- Transfer: configure send protocols, send and receive files, cancel transfers
- Macro: run, edit or record macros, wait for a selection, encrypt strings or emissions
- Options: use firewall (proxy), start Tun accessories, change interface language
- Window: Arrange windows
- Help: Online help and About dialog box

Displaying the menus

Regardless of the menus available, you can hide or show the menu bar in the emulator. If you hide the menu bar, it's the same as displaying the emulator in full screen mode without menus like a dumb terminal.

To show or hide the menu bar, choose Options from the Settings menu. If you've already hidden the menu bar, you can access this option by right-clicking anywhere on the screen.

If you want to apply the change immediately, click the Customize tab and select or clear the Display session in full screen without menus check box.

If you want to change the emulator's display for the next time it's started, click the General tab and choose the option you want.

Example:

On startup, the emulator appears in its default size with the menu bar visible. You want to change this so the emulator starts in full screen mode without the menu bar. Choose Options from the Settings menu and then choose Full Screen Display (No menu) on the General tab. Quit the emulator and restart it: It starts in full screen mode without menus.

Note:

Integrators of emulation solutions can offer their users a customized asynchronous emulator with menus and menu options they have chosen themselves. See “Emulation customizer” on page 98.

Toolbar

The toolbar lets you access simply and rapidly the emulator's main commands.

By default, the emulator displays a toolbar containing some of the configured buttons.

You can:

- Display or hide the toolbar (Toolbar option in the Settings menu).
- Display small or large buttons (Options in the Settings menu, Customize tab, Small Toolbar Buttons check box).

You can customize the toolbar:

- Floating toolbar and docking: To do this, left-click the gripper (at the end of the toolbar) and move the toolbar with the left mouse button held down. When the mouse pointer reaches one of the edges of the emulation session, the toolbar will dock automatically.

Toolbar buttons and separators: Display the toolbar context menu (right-click the toolbar) and choose Customization.... Add, delete and classify the buttons or separators you want. Click Reset to return to the default toolbar.

Toolbar style: Can be flat (default) or not (choose Options from the Settings menu, Customize tab, Flat style).

You can also choose your own toolbar bitmap that you can customize with an appropriate application. Specify the bitmap path in the Registry setting BitmapPath in the key:

```
HKEY_LOCAL_MACHINE/Software/Esker/Tun/8.00/TunEmul/Toolbar
```

Also in the Registry, you can change the order of the buttons (Order) and the number of buttons in the toolbar (BitmapButtons).

Status bar

Choose Settings > Status Bar to show or hide the status bar.

The status bar contains the following information:

- Current session information.
- Communication mode used.
- Cursor coordinates.
- Print job indicator.
- Keyboard status indicator that you can click to lock or unlock the keyboard.
- Caps, Num and Scroll Lock indicators.

Emulation over TCP/IP

To open a new emulation session on a TCP/IP network, choose New Session and then TCP/IP... from the File menu.

Settings

Fill out the fields required for connection, following the instructions below, then click on OK.

Connection type

If the connection type displayed doesn't correspond to the one you want, you can select another connection between the following three types:

- Connection via TCP/IP network (Telnet option)
- Connection via RS232 link (Serial option)
- Connection via modem (Serial with modem option).
- Connection via SSH (SSH option)

Depending on the type of connection chosen, various fields will appear.

Terminal type

Select the type of terminal you want to emulate from the Terminal Type drop-down list. The terminal type you choose depends on the server type or the type of application you want to use.

> See "Terminals" on page 14.

Display settings

In the Display settings field, select all the parameters defining the session working environment: font to be used, screen dimensions, font color and style, screen background, function key-panel, use of the mouse, etc. This field is optional.

Host name (TCP/IP connection)

Enter the name or IP address of the host you want to connect to in this field (only enter a name if you have a name server).

The list contains the servers registered in the hosttab file.



If you use an NIS server, all the servers referenced on it appear in the list.

Telnet port number (connexion TCP/IP)

By default, the port number corresponding to the Telnet protocol is 23. If your configuration uses another port, enter the new value in this field.

Starting an emulation session

After selecting the host and terminal type, click OK to open the connection. If TCP/IP is running correctly on the PC and host, you should receive a login prompt.

Emulation over serial link

The emulator can use serial ports COM1 through COM32 simultaneously to connect to several different hosts

To open a new emulation session over an RS232 asynchronous link, choose New Session from the File menu and then RS232....

Settings

The Connection type, Terminal type and Display settings fields can be filled just like a TCP/IP connection. See Emulation session on a TCP/IP network for more details.

Serial Port

Select the PC communication port you want to use from the drop-down list (COM1 through COM32).

Baud Rate

The transmission speed over an asynchronous line is measured in bauds (bits per second). The baud rate can be set between 75 and 115200. The value should match the speed set on the server (in /etc/gettydefs).

Data bits

This field describes the number of significant bits used to make a byte. This number is almost always either 7 or 8. Check the setting on your host machine to be sure.

Stop bits

Either 1 or 2 bits mark the end of a byte.

Parity

The parity bit provides protection against transmission errors. The parity bit can be either even or odd. None means that no parity bit is sent after a byte.

In 8-bit transmission, Space or Mark can be used to set the last bit in a byte to 0 or 1 respectively.

Advanced configuration (RS232 or modem connection)

Click on the Advanced button to complete configuration of your connection.

I/O buffer size

This field defines the size of the Input/Output buffers (in bytes). The default value of 2048 is usually sufficient.

Flow control

Flow control keeps the Input/Output buffers from overflowing and losing data. It's very important that Tun EMUL is set the same as the host machine.

The Xon/Xoff and Xany/Xoff flow controls are the most commonly used on UNIX servers.

With Xon/Xoff, when the buffers in the PC in emulation are 75% full, Tun EMUL sends a DC3 (^S) character to the host asking it to suspend data transmission. When the buffers are 75% empty, Tun EMUL sends a DC1 (^Q) character to request that the host resume data transmission.

When using Xany/Xoff, the PC in emulation still sends a DC3 to suspend transmission, but can send any character to resume.

Some UNIX servers handle flow control directly through cabling. Instead of using special characters (DC1 & DC3), electronic signals are sent when the PC's buffers are full. This is known as hardware handshaking. In general, two types of hardware handshaking are used:

- DTR and DSR signals
- RTS and CTS signals

Select the values you want with the mouse.

Starting emulation

Click OK when you've set the above parameters to connect to the host.

Serial connections are more difficult than TCP/IP connections, and may require special attention to cabling and host configuration. See "Serial Cabling" on page 176.

Emulation over TAPI Modem

Tun uses the TAPI standard interface to configure your modem settings. This interface has the following advantages:

- A simple modem installation procedure, thanks to the automatic detection.
- A single configuration. Once configured, the modem can be used by other communication applications.

The TAPI modem connection specific settings are described below.

Modem

This list contains all the modems installed on your PC.

Install and configure your modem using the Modems icon in the Windows Control Panel. This dialog box enables you to easily install your modem thanks to the automatic detection.

- > For more details on the fields of this box, refer to the Communications / Modems and communication tools / Setting up a modem section of the Windows Resource Kit Help. If this online help is not installed on your PC, you can find this file on the Windows Installation CD-ROM.

Use the Configure the modem button to display the selected modems settings in the list. The dialog box that appears is the same one that you use to configure your modem from the Modems icon in the Control Panel.

Phone number

You can enter the telephone number of the host system in this field, unless you prefer to dial manually.

Commas (,) can be used to insert a two-second pause during dialing. You could find this necessary for obtaining external lines in office buildings or for dialing internationally (for example, 011,33,472834646).

If you complete this field, the emulator will try to dial up over the modem as soon as the session opens.

Timeout

If no response is heard from the remote host in the time you specify, the emulator considers the connection as failed. Thirty seconds is usually a reasonable value.

Dialing rules

Select the Use dialing rules option if you want to automatically modify the phone number dialled according to the geographic area called or the calling area.

Click the Dialing rules button to configure the dial settings.

Country code

Select the country you want to dial in the list. The selected country code will precede the server phone number to form the dialled number.

Area code

Specify the area code called (optional). This code is used in specific countries only.

Call location

In the drop-down list, select the configuration that corresponds to your calling area. A configuration contains several parameters that simplify dialling according to your phone settings. To change these parameters, click Advanced. The dialog box that appears can also be displayed from the Modems icon in the Windows Control Panel, using the Dialing properties button. Refer the Windows Help for more details on the contents of this dialog box.

Emulation over SSH

Tun uses the OpenSSH standard to provide SSH connectivity. To open a new emulation session using SSH, select File>New Session>SSH.

Settings

SSH Port Number: This defaults to the SSH standard of 22, but can be modified if your host requires a different port for SSH connections.

SSH User Name: The SSH account name used for logging onto the host.

SSH Properties: Click the SSH Properties button to configure SSH. The SSH Properties dialog will open.

SSH General tab

Enable SSH 1: Select this option to activate SSH protocol version 1 for this session. This option is enabled by default. Protocol version 1 permits the use of Triple DES (3DES) and Blowfish encryption ciphers.

Enable SSH 2: Select this option to activate SSH protocol version 2 (the newer version) for this session. This option is enabled by default. Protocol version 2 permits the use of Arcfour encryption as well as Triple DES (3DES), Blowfish and CAST128 ciphers in CBC mode.

Note: If both SSH versions are checked, the highest SSH version supported by the host will be used.

Enable compression: Allows the client to compress data before encryption. If the host also supports compression, then transmissions between the client and host are compressed. For large files, this decreases transmission times over slower connections.

SSH Host Authentication tab

There are three locations for storing host authentication keys. All three locations will accept both SSH1 and SSH2 level keys. You can recognize each type of key by its

format. SSH1 keys contain the host name or IP address, the RSA key length, the decimal exponent and modulus, followed by optional comments. The SSH2 keys contain the host name of IP address, the type of key ("ssh-rsa" or "ssh-dss"), the base 64 encoded public key, followed by optional comments.

Known Hosts: Copy and paste host keys directly into the box. Hosts listed here are treated as known hosts for this workspace, and are eligible for public key authentication. This box accepts both SSH1 and SSH2 level keys.

Use system known hosts file: Select this option to use the system known hosts file. Click Edit to add, change, or delete keys from this file. The system known hosts file is set up by the administrator, and lists known hosts that can be used by all users on a particular machine.

Use user known hosts file: Select this option to use the user known hosts file. Click Edit to add, change, or delete keys from this file. The user known hosts file lists known hosts that can be used by a specific user on a particular machine.

If you disable this option, you will not be able to save new host keys, and you will see a warning message each time you try to log on to an unknown host, even if you have logged on to it before.

Accept unknown hosts: Allows you to connect to hosts that have not been saved as known hosts. Once you have connected to a host this way, Tun asks if you want to add that host's key to the user known hosts file, if using that file is enabled.

Ask for confirmation for unknown hosts: If this option is enabled, a message appears whenever you try to connect to a host that has not been saved as a known host. Once you have confirmed that you want to connect to an unknown host, your connection to the host is established.

Reset defaults: Click this button to change each setting on this tab back to its default value. The defaults are based on the current Open SSH standard at the time this version of Tun was developed.

SSH System or User Known Hosts File dialogs

These dialogs are accessed by clicking the Edit buttons in the SSH Host Authentication tab. They allow direct access to the known hosts files.

Known Hosts: Copy and paste host keys directly into the box. Hosts listed here will be treated as known hosts and eligible for public key authentication. The keys here will be saved in the file listed below. This box accepts both SSH1 and SSH2 level keys. This box will be grayed out if you do not have write access to the file.

File path: Shows the location of the file containing the host keys listed in the text box above.

Note: If the user has read only access to the file, the File path and Known Hosts boxes are uneditable.

SSH User Authentication tab

On this tab, you can generate public/private key pairs. Once you generate the key pair, copy the public key to the host for use in key authentication.

Generate keys: Click these buttons to generate a user authentication key pair (public and private) described below. These may be used to authenticate the user on the server machine.

RSA public key: Lists the public key generated using the RSA algorithm. This key is used with the SSH 2 protocol.

DSA public key: Lists the public key generated using the DSA algorithm. This key is used with the SSH 2 protocol.

SSH1 public key: Lists the public key generated using the RSA (from SSH1) algorithm. This key is used with the SSH 1 protocol.

Note

: To delete a key, you must delete the key file from the appropriate directory:

Key files

RSA keys: rd_isa and rd_isa.pub

DSA keys: id_dsa and id_dsa.pub

SSH1 keys: identity and identity.pub

Key file directories

Windows 2000/2003 and XP: C:\Documents and Settings\\Application Data\Esker\SSHX

Key length: Sets the size of generated keys in bits. The value must be between 0 and 9999; the default is 1024. Note that entering a value lower than 1024 could result in compromised security. Entering a very large value can significantly extend the time it takes to generate the key.

Note: After changing the key length, click Apply before generating a new key or the program will use the previous key length.

SSH Advanced tab

To change the values, simply add, change, or delete the text in each box. The items in each box on this tab are comma-separated.

Public key algorithms: Lists the algorithms supported by the client.

Ciphers: Lists the ciphers supported by the client.

MACs: Lists the MACs supported by the client.

Authentication methods: Lists the authentication methods supported by the client.

Reset defaults: Click this button to change each setting on this tab back to its default value. The defaults are based on the current Open SSH standard at the time this version of Tun was developed.

Sessions and configurations

Managing multiple sessions

As the asynchronous emulator is an MDI application (Multiple Document Interface), you can open several connections simultaneously on the same server or different servers.

To open a new session, select File > New Session and select the type of connection you want (TCP/IP, RS232, etc.). For more details, refer to one of the preceding sections on emulation session settings (for example, Emulation over TCP/IP to open a session in a TCP/IP network).

Each session is displayed in a separate window in the emulator. You can arrange the windows using the mouse or the Window menu options.

Use the keyboard shortcuts <Alt><F1>, <Alt><F2>, <Alt><F3> and <Alt><F4> to quickly access the first four sessions.

Configurations

A configuration refers to all the settings used in one or more sessions: Terminal settings, display settings, macros, function-key panel, etc. All these settings are contained in the same configuration file: That way, the exact same working environment can be reproduced at a later time.

For example, a configuration can be the combination of a TCP/IP session on an ANSI terminal with display settings A (font, colors, background, function-key panel) and a TCP/IP session on an hterm terminal with display settings B.

Esker's asynchronous emulator can collect all the settings files or session files (terminal files, keyboard files, escape sequence files, macro files, function-key panel files, etc.) together in an archive file with the extension .cfz. The archive file lets you share the settings files easily with other users. You can distribute the archive on diskette, for example, or via an NIS server or TFTP, etc.

For upward compatibility reasons, you can load .cfg extension configurations in the synchronous emulator. In this type of file, the settings related to the session(s) configuration(s) are simply memorized in a .cfg extension file. This file specifies which files are used for the configuration session(s). However, any new configuration will be saved in a .cfz archive file only.

Note:

The difference between a .cfg configuration file and a .cfz archive file is that the configuration file only contains references to the other files in the configuration while the archive file is a file container that can be distributed to other users (like a .zip file). A .cfg configuration file is of little use by itself.


Saving a configuration in an archive file


To create or update an archive containing all the files associated with the configuration, choose File > Save....

Select the archive file or enter its name (extension .cfz).

Opening a configuration

To use an existing configuration, do one of the following:

Choose File > Open Configuration... or click the button . Select the configuration file you want (a .cfg configuration or a .cfz archive).

Choose Settings > Configuration... or click the button . Then click Load and choose the configuration file you want (a .cfg configuration or a .cfz archive).



Click NIS if you want to load a file from the NIS.

You can also run the emulator using the configuration name as a parameter as follows:

```
C:\Program Files (x86)\Tun\Emul\emul32 demo.cfm
```

Modifying the contents of a .cfz archive

To modify an archive's contents, select Settings > Configuration.... The appearing dialog box contains the settings of the currently opened sessions.

Select the session for which you want to modify the archive, and click Edit CFZ to open the following resources editor.

Resources editor


The resources editor is used to associate with the standard session and resources any additional resources the session may be using.. The standard resources are those files directly associated with the session when it is configured. They include:

- The terminal file (file extension .ter).
- The customisation files (keyboard, function keys, active zones, colours, screen background image, etc.).
- The key panel file where applicable (file extension .pan).
- The start and end macro files where applicable.

The additional resources are, for example:

- A second key panel file used in place of the standard key panel file (and displayed when executing an action designed to change key panels).
- A macro file called up by an action triggered when a key is pressed.

The resources editor also enables standard resources to be included in the session archive file (workspace .cwz): the session is established independently, without looking for its configuration files on the disk. In this case, even if they are moved, modified or deleted from their location, the session will be established using the parameters previously archived.

To add an extra resource (for example, a key panel file called up by an action), click on the  button and select the file to be added.

To delete a resource from the list, select it and then click on the  button.

In order also to archive the standard resources to make the session independent, select the Generate an self-sufficient configuration at saving check-box.

Editing and saving .cfg configurations

To create a new session in your configuration, choose Settings > Configuration.... The dialog box displayed shows the settings of currently opened sessions.

You can add a new session to the configuration list, or delete and modify sessions. Click Add, Remove or Change to do that, and add settings as explained above in the section on sessions.

To add a saved configuration, click Load....

To save the session settings in the list, click Save (or Save As... if you want to change the name or location of an existing configuration).

Select the file or enter its name (extension .cfg) and click Save.

Note:

To save a configuration in an archive file, you can choose File > Save As.... The configuration saved consists of all the open sessions with their respective settings.

Terminal display and revamping

Esker's asynchronous emulator lets you customize the look of the opened sessions. The relevant settings are assembled in display settings files.

Display settings files specify:

- Character font size, type and attributes
- Colors and effects
- Background
- Size of the emulation screen
- Function-key panels
- Customizable toolbar
- Hostpot activation
- Scroll bars

You can save display settings files and attach them to session configurations (described in the preceding section). These .ctx files work with both the Windows and MS-DOS emulation modules.

The powerful facilities offered by the configuration process give a graphic look and feel to traditional UNIX applications and amount in practical terms to a revamping of UNIX applications.

Click Settings > Display Settings... to select the elements you want to use.

Loading /Saving .ctx files

If you want to load the settings defined in an existing display settings file, click Load and select the .ctx file you want.

If you want to save new or changed settings, click Save and enter the name of the .ctx file.

To associate display settings with an emulation session, first check the settings defined for the session will be saved when the session is saved. See “Emulator options” on page 28. Then save the session to a configuration file. See “Sessions and configurations” on page 25.

See “Customizing the emulator” on page 64 to fill out all of the context parameters.

Emulator options

To complete your customization of the asynchronous emulator, you can set different options for the emulator working environment.

To do that, choose Options from the Settings menu.

Startup and disconnection options

Choose Settings > Options from the main menu to set the screen options, customize the toolbar and select a keyboard. On the General tab, choose the startup and disconnection options you want.

Select the size of the emulator startup screen: The options are maximum size, the size used during the previous emulation session, icon mode, and default size. The default size is set on the Terminal Font tab of the Display Settings dialog box (Settings > Display Settings...). You can also choose full screen display without menus.

Choose the default screen size for the opening of a new session.

Choose the Save options for any changes you make: Never to never save the changes you make (for example, to the display settings) when you quit the emulator (or close a session), Always to always save your changes automatically, or Save changes on confirmation if you want Tun EMUL to display a confirmation dialog box each time you quit the program or close a session. The last option is the default.

Select the session termination option you want, with or without confirmation.

If you chose the default size for the emulation session (see above), you can change the window coordinates and save your changes with the option Save window coordinates. Save your session settings in a configuration file. That way, when you re-open the configuration, the session window will look as you set it.

National keyboards

On the National Keyboard tab, select the keyboard language for your system. For more details, see “Advanced Use of Asynchronous Emulator” on page 137.

Customizing the asynchronous emulator

On the Customize tab, you can set the following options:

Select the Display session in full screen without menus check box, if you want to switch to full screen mode immediately. In full screen mode, you can right-click anywhere on the screen to display a context menu containing the main menus and options.

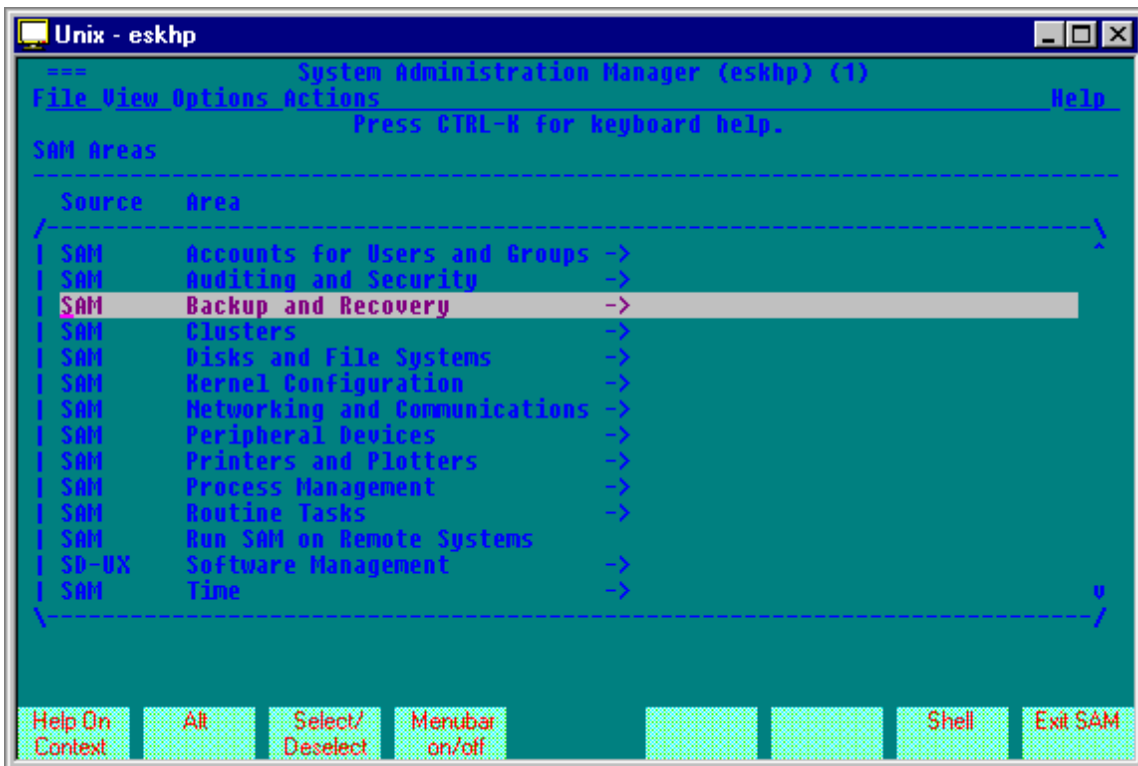
Choose whether to display small buttons on the toolbar or not. This is useful if you customize the toolbar with a lot of extra buttons (you can click an empty space in the toolbar with the right mouse button and then Customize... to do this).

Select or unselect the Flat toolbar style to change toolbar style.

Click the ... button to customize the toolbar .

HP emulation characteristics

The HP emulation provided by Esker fully integrates the Setup menu available on a HP 700/98 terminal: commands appear in the lower part of the emulation screen. They can be replaced with hotspots.



The Setup menu commands are managed by escape sequences, each application having its own commands.

Two modes are available: the user mode and the system mode. As with a standard HP terminal, you can switch modes using the F9 (also used to mask Setup menu) and F10 keys.

The HP emulation also integrates:

- Typed fields management: alphabetic, alphanumeric, integer, signed decimal, filled integer, filled decimal...
- Areas management: using Tab, you can move the cursor from field to field within a given area, instead of moving from line to line.

Dynamic Data Exchange

DDE is a standard inter-application communication protocol defined by Microsoft. It allows Windows applications that support this protocol to exchange data with each other.

Two applications that take part in a dynamic data exchange are said to be engaged in a DDE conversation. The application that starts the DDE conversation and that wants to access data is called the DDE client. The application that answers the client and that has access to the data and can transfer it during the DDE conversation is called the DDE server. An application can take part in several conversations at the same time: It can behave as a client in some conversations and a server in others.

To allow the exchange of data between the client and server applications, the data must be identified by an application name, a topic name and an item name. At the start of the DDE conversation, client and server determine the names of the application and topic. Next, a particular item of data to exchange is specified.

Application

This is the name of the server application to which the client sends its data requests. Application names are sometimes known as service names.

Topic

A topic is a group of items that can be used in a DDE conversation. For example, with applications that manage documents as files, a topic is typically a file.

The system topic is a special topic that supplies a group of server application data items that can be widely used by other applications. This topic is always available when the server application is run.

Item

An item is a unit of DDE data linked to the topic that is changed between applications during the conversation. For example, this could be an individual cell in a spreadsheet.

Conversations

One of the powerful features of DDE is that a client application can send commands or submit requests to a server application that has been designed to recognize and accept them. The type of command or request that a server can accept depends on the server.

Put simply, a DDE communication between a client application and a server application is like a phone conversation:

The client begins by initiating the conversation using a function like DDEInitiate which defines a DDE link. This function typically contains an application name, a topic and possibly an item that function like a phone number since they let the client identify the server with which it wants to communicate.

Once the link is defined, and a channel number attributed to it, the client can use this reference number to:

Send data items to the server using a function like DDEPoke (the name may change depending of the application)

Fetch data items from the server using a function like DDERequest (the name may change depending of the application)

Send commands to the server using a function like DDEExecute (the name may change depending of the application)

Finally, when the client has finished communicating with the server, it hangs up by ending the DDE link using a function called DDETerminate (or something similar). Note that closing the server or client application always ends the links between the two.

Tun EMUL and DDE

Tun EMUL is a DDE server. It uses a link between a Windows application and the emulation screen to let the application benefit dynamically from data on a server that's accessed through an emulation session.



In practical terms, you use DDE when you copy a part of the emulation screen (Copy option in the Edit menu) and paste it with a dynamic link (Paste Special option in the Edit menu) into a Windows application (like Word or Excel, etc.). The dynamic link created using the Paste Special command automatically updates the Windows application document if the original data changes. One use of this technique is the updating of graphs. If you link an Excel document containing a graph to a database via the emulator, the graph is

updated automatically as the data changes in the database. You can also use DDE in Windows applications you develop yourself.

Use

Emulwin is used as the server application name. In Tun EMUL, a topic corresponds to an open session. Each topic has its own name. The topic name is the name of the host. If more than one session is opened on the same host machine, the topic name, like the session name, is the name of the host followed by a colon (:) and a number.

For example, the first session opened on the host machine *risc* is called *risc*, the second *risc:1* and the third *risc:2*. The system topic is called *System*. Application names associated with topic names are used to start and end DDE exchanges (initiate and terminate).

The contents and structure of topic items, as well as the commands, depend on the server application. Server and client can exchange recognized items (using *poke* and *request*) and the client can send supported commands to the server (using *execute*). The items recognized by the Tun EMUL DDE server for its standard system topic and the other topics are described below.

Note:

Tun EMUL supports the Execute Control 1 protocol which returns information in reply to a command request from a client application. Tun EMUL uses this information to supply an error message when a command fails. For example, you can declare an item name, recognized by the topic, using the command `[Result(name)]`. This item contains the results of the commands that are subsequently executed. If a command fails on execution, the error message returned by the server is placed in `name`.

Items and commands supported by the system topic

Items:

- `SysItems`: Returns a list of system topic items.
- `Topics`: Returns a list of available topics, that is the sessions opened in Tun EMUL.
- `Formats`: Returns a list of supported formats.
- `Protocols`: Returns a list of supported protocols.
- `Help`: Returns a help on the use of the DDE server.

Commands:

- `Open (configurationfilename)`: Opens a new session (that is, one or more pre-defined sessions); an error occurs if Tun EMUL can't find the configuration file, if the file contains errors, or if a configuration file is already open.
- `Close`: Closes all the emulation sessions.
- `Resize(0)`: Reduces the size of the server application window.
- `Resize(1)`: Returns the window to its original size (before it was reduced or increased).
- `Resize(2)`: Increases the size of the server application window.

- `Result(name)`: Defines the item that contains the message returned by executed commands.

Items and commands supported by the other topics (emulation sessions)

Items:

- `TopicItemList`: Returns a list of the topic's items.
- `Formats`: Returns the formats supported by the topic.
- `Screen (page, line, columns, length)`: Returns a selected string of given length.
- `ScreenRect (page, start line, start column, end line, end column)`: Returns a rectangular selection.
- `Host`: Sends data to the host machine.

Commands:

- `SendData(string)`: Sends a string over the connection. The data can't be sent if a file transfer is taking place.
- `Macro(name, arg)`: Executes the macro name, passing it the optional parameter `arg`: An error occurs if the macro is already running or if a file transfer is taking place.
- `Terminate`: Closes the emulation session. The session isn't closed if a file transfer is taking place.
- `Result(name)`: Defines the item that holds the messages returned by the executed commands.

Command syntax

Commands must respect the following syntax:

```
[command(arg, ...)]
```

You can pass command arguments directly as long as they only include the following characters: a-z, A-Z, 0-9, _ and \$.

To pass a more complex character string as an argument (including, for example, spaces and slashes), put it in quotes (inverted commas) and follow these instructions:

To pass the quotation marks character as an argument, you must double it: So the command `[command("")]` passes single quotation marks as an argument.

To pass the backslash character `\`, you must double it, otherwise it's ignored: For example, the command `[command(x)]` passes only the character `x`; you must use `[command(\\)]` to pass the character `\`.

The command `SendData(string)` also uses special Tun EMUL encoding in addition to the DDE encoding:

| Notation | Meaning |
|-----------------|--|
| \e | Escape character |
| \E | Escape character |
| \n | Line feed |
| \r | Carriage return |
| \t | Tab |
| \b | Backspace |
| \f | Form feed |
| \s | Space |
| \\ | Backslash “\” |
| \0xn | Character’s hexadecimal value (n: 0 through FF) |
| \0n | Character’s octal value (n: 0 through 377) |
| \n | Character’s decimal value (n: 0 through 255) |

For example, to pass a single backslash character (\) as an argument of SendData, use the Tun EMUL notation \\, doubling the backslash: You must then double these backslashes for DDE decoding to give [SendData(\\)]. To pass a line feed character, you must use the Emul notation \n, and double the backslash character: [SendData(\\n)].

Examples

You can use the topics, items and commands described above in application programming. The following examples can be used in Word (WordBasic) and Excel (Visual Basic for Excel).

Initializing a DDE conversation

| | |
|-------|---|
| Word | DDEInitiate(Application\$, Topic\$) ex: channum=DDEInitiate(Emulwin, System) |
| Excel | object.DDEInitiate(app, topic) ex: channum=Application.DDEInitiate(app:="Emulwin",topic:="System") |

Sending data

| | |
|-------|--|
| Word | DDEPoke Channel, Item\$, Data\$ ex: data\$=DDERequest\$(channum, "ScreenRect(0,8,41,12,43)") |
| Excel | objet.DDEPoke(channel, item, data) ex: Application.DDEPoke channum, "Host", "text" |

Retrieving data

| | |
|-------|---|
| Word | DDERequest\$(Channel, Item\$) ex: DDEPoke channum, "Host", "text" |
| Excel | objet.DDERequest(channel, item) ex: data=Application.DDERequest(channum, "ScreenRect(0,8,41,12,43)") |

Sending characters

| | |
|-------|--|
| Word | DDEExecute Channel, Command\$ ex: DDEExecute channum, "[SendData("+Chr\$34+"text"+Chr\$34 +")]" |
| Excel | objet.DDEExecute (channel, string) ex: Application.DDEExecute channum, "[SendData("+Chr\$34 + "text" + Chr\$34+")]" |

Closing the conversation

| | |
|-------|--|
| Word | DDETerminate Channel |
| Excel | object.DDETerminate(channel) ex: Application.DDETerminate channum |

Example

This example illustrates an alternative method to Paste with link to regularly update data in an Excel spreadsheet from a UNIX server. It implements DDE between an Tun EMUL emulation session (DDE server) and an Excel macro (DDE client).

Tun EMUL includes a DDE example that uses a DDE link between data on a UNIX host and an Excel spreadsheet running a macro. To view the results, do the following:

Copy the file ddedemo2.sh to the /home directory on the UNIX server from the directory C:\...\Tun\Emul\Demo\Dde.

In Tun EMUL, create a configuration in which the first session is a link to your UNIX server and save it as ddeconf.cfg in the directory C:\...\Tun\Emul\Demo\Dde.

Start Excel and open the file ddedemo2.xls in the directory C:\...\Tun\Emul\Demo\Dde.

Open the spreadsheet Chart1 or Sales to view the results.

To run the macro, choose Macro from the Tools menu, select the macro DDEMacro and click Run, or else use the keyboard accelerator CTRL-D.

Note:

The macro refers to C:\Program Files (x86)\Tun\EMUL\emul32.exe in its code, which means that your application should be in the to C:\Program Files (x86)\Tun\EMUL directory. Otherwise, you should enter the path corresponding to your application directory in the ddedemo2.xls macro.

The macro uses the following Visual Basic for Excel DDE commands:

object.DDEInitiate(app, topic):

app: DDE application server name

topic: topic name

This function returns the number of the channel opened for the DDE link.

object.DDERequest(channel, item):

channel: channel number returned by the DDEInitiate method

item: item requested

object.DDEExecute(channel, string):

channel: channel number returned by the DDEInitiate method

string: command defined in the receiving application

object.DDETerminate(channel):

channel: channel number returned by the DDEInitiate method

In all the above methods, the object is optional and refers to the Application object (Microsoft Excel).

This macro is written in Visual Basic for Excel. It performs the following operations:

1. Starts Tun EMUL.
2. Initializes a DDE link between an Excel spreadsheet and Tun EMUL's system topic using the method DDEInitiate.
3. Opens a specific configuration using the method DDEExecute and the command Open, and reduces Tun EMUL's main window with the command Resize.
4. Fetches a list of open sessions using the method DDERequest with the item Topics, and selects the first session from the list (corresponding to the UNIX server).
5. Initializes a DDE link between the Excel spreadsheet and this session using the method DDEInitiate.
6. Starts the emulation session by executing an Tun EMUL connection session and sending commands using the method DDEExecute with the command SendData.
7. Updates the data from the emulation screen using the method DDERequest with the item ScreenRect.
8. Closes all the emulation sessions using the method DDEExecute with the Close command from the system topic.
9. Closes the open DDE links using the method DDETerminate.

The code for the sample macro is as follows:

```
Sub DDEMacro()  
  
    RetVal = Shell("C:\PROGRA~1\TUN\EMUL\EMUL32.EXE", 1)
```

```
canal1 = Application.DDEInitiate(app:="EMULWIN", topic:="System")
Application.DDEExecute canal1, "[Open(" + Chr$(34) + "demo\\\\dd-
e\\\\ddeconf.cfg") [Resize(0)]"
listTopics = Application.DDERequest(canal1, "Topics")

session1$ = listTopics(1)

waitTime = TimeSerial(Hour(Now()), Minute(Now()), Second(Now()) +
3)
Application.Wait waitTime

canal2 = Application.DDEInitiate(app:="EMULWIN", topic:=s-
ession1$)
DDEExecute canal2, "[Macro(" + Chr$(34) + "demo\\\\d-
de\\\\logindde.mac + Chr$(34) + ")]"
waitTime = TimeSerial(Hour(Now()), Minute(Now()), Second(Now()) +
20)
Application.Wait waitTime

DDEExecute canal2, "[Senddata(" + Chr$(34) + "cd home" + "\\r" +
Chr$(34) + ")]"
DDEExecute canal2, "[Senddata(" + Chr$(34) + "./ddedemo2.sh" +
"\\r" + Chr$(34) + ")]"

For k = 1 To 20

    waitTime = TimeSerial(Hour(Now()), Minute(Now()), Second(Now
()) + 1)
    Application.Wait waitTime

    sales = DDERequest(canal2, "ScreenRect(0,8,41,12,43)")

    For i = LBound(sales) To UBound(sales)

        s$ = sales(i, 1)

        j = i + 4

        cell$ = "D" + j

        Worksheets("Sales").Range(cell$).Value = Val(s$)

    Next i

Next k

Application.DDEExecute canal1, "[Close]"

Application.DDETerminate canal1
```

```
Application.DDETerminate canal2
```

```
End Sub
```

Note:

The TimeSerial function and the Wait method are used in the macro code to synchronize it with Tun EMUL. The synchronization timeout value is:

```
waitTime = TimeSerial(Hour(Now()), Minute(Now()), Second(Now()) + 1)
```

If necessary, you can change the value by increasing the number of seconds added to Second(Now()).

IBM Synchronous Emulation

Synchronous terminal emulation provides access to the IBM MainFrame (3270 emulation) and AS/400 (5250 emulation) servers.

Characters no longer flow freely between terminals and servers as in asynchronous emulation, but rather circulate in packets. With this type of communication, terminals don't send data until the server has responded to the previous request, and vice versa. Data exchange between the two machines is structured and organized.

Typically, synchronous emulation is less flexible than asynchronous emulation, and offers fewer possibilities for customization and personalization. Esker's synchronous emulators, however, have powerful revamping functions that let you define a more attractive environment.

The main advantages to synchronous emulations is that they are perfectly adapted to the transactional type of architecture (the forerunner of client/server), and offer excellent performance in packet-mode networks such as X.25 and X.32.

Connection protocols

IBM mainframe servers transmit and receive the 3270 type of data flow. To communicate with this type of server, the terminal emulator can establish a connection based on the following protocols: :

- Connection using the Telnet 3270 protocol (TN3270) over TCP/IP.
- Connection based on the extended version of this protocol: TN3270E.
- Connection via a UNIX SNA-TCP/IP gateway in TN3270 or TN3270E mode.
- Native mode connection via a Netware for SAA gateway.
- Native mode connection via a Microsoft SNA Server gateway (LUA or FMI).

In the IBM environment, IBM AS/400 servers send and receive 5250-type data flow. To establish a connection with this type of server, the PC can establish the following types of connection:

- Connection using the Telnet 5250 protocol (TN5250) over TCP/IP.
- Connection based on the extended version of this protocol: TN5250E.
- Connection through a UNIX SNA-TCP/IP gateway in the TN5250 or TN5250E mode.
- Native mode connection through a Netware for SAA gateway over TCP/IP or IPX/SPX.

Native mode connection through a Microsoft SNA Server gateway over TCP/IP or IPX/SPX.

Note:

The TN3270E and TN5250E protocols allows the 3287 and 3812 printers emulation. See the 3287 and 3812 printer servers chapter.

TN3270E Emulation

TN3270E emulation is 3270 emulation enhanced by the addition of two main features (RFC 1647):

- Choice of LU (logical unit): An LU is an IBM concept that's similar to UNIX devices. For example, a printer that's connected via telnet to an IBM Mainframe always uses the same LU. One or more LUs are often linked to a user profile. This allows access to menus, depending on the LU used. TN3270E lets you select a specific LU according to the processing you want to perform.
- Extra function-key: SYSREQ lets you interrupt an application to return to the system menu.

Esker's synchronous emulator offers TN3270E-type emulation, integrating the extra features (as well as the ATTN function-key which interrupts a process at the telnet stage).

TN5250E Emulation

TN5250E emulation is 5250 emulation enhanced by the addition of features (RFC 1205):

- Choice of the AS/400 system name (as defined on the AS/400).
- Configuration of options that enable automatic connection to the AS/400 system, eliminating the need to access through the standard login page.
- Configuration of advanced options that enable the modification of settings related to the AS/400 system name.

Esker's synchronous emulator offers TN5250E-type emulation, integrating the extra features.

3270 gateways

The gradual adaptation of SNA communication protocols (IBM) to other protocols like TCP/IP (Internet) has created a need for SNA gateways.

The Esker IBM 3270 emulator lets you connect to MainFrame servers directly via a TCP/IP network, using the TN3270 or TN3270E protocols, or via SNA gateways:

- UNIX SNA–TCP/IP gateways in TN3270 or TN3270E mode.
- Netware for SAA gateways in native mode (specific to the gateway).
- SNA Server (LUA or FMI) gateways in native mode (specific to the gateway) .

Note:

The native mode offers the same functions as TN3270E (choice of LU, SYSREQ) with the added advantages that:

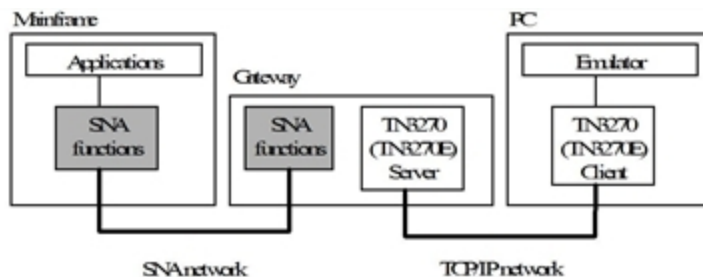
--It lets the gateway work with different network protocols: TCP/IP, IPX and NetBeui.

--It dynamically manages the gateways used to access the mainframe, if there are more than one. The client application (Esker's 3270 IBM emulator) need only send a connection request over the network for it to be answered by whichever of the gateways is available.

UNIX SNA-TCP/IP gateways

Esker's **3270** IBM emulator lets you connect to SNA-TCP/IP gateways on UNIX machines in TN3270 or TN3270E mode only (native mode isn't supported).

The PC emulates a 3270 terminal (or 3270E) by connecting to the gateway as if it was a server.

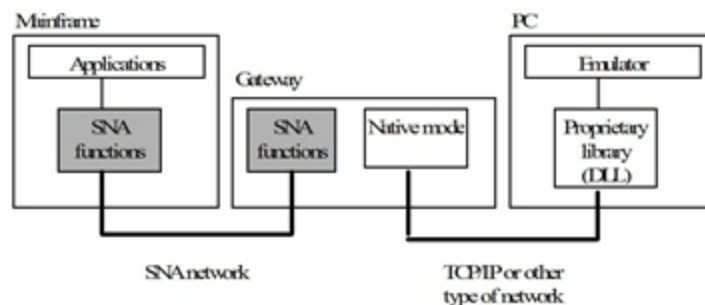


You can use UNIX gateways like Open Connect's (OC/TELNET server), Apertus's (Express SNA server), TPS System's (TPS®/SNA server), and Seasoft's (HAL3270 server).

Novell Netware for SAA gateways

Esker's IBM 3270 emulator lets you connect to Netware for SAA gateways in native mode.

The emulator communicates with the Winrui32.dll library located on the PC and supplied by Novell. This DLL then manages the connection with the gateway:



Netware for SAA gateway prerequisites

To use Esker's IBM 3270 emulator via a Netware for SAA gateway in native mode, you must first install the Novell Netware client on your PC (which includes dll Winrui32.dll) as well as the IntranetWare for SAA client from Novell.

The connection parameters are grouped together in the 3270 emulator. Make sure the Novell server is correctly configured before connecting with the 3270 emulator.

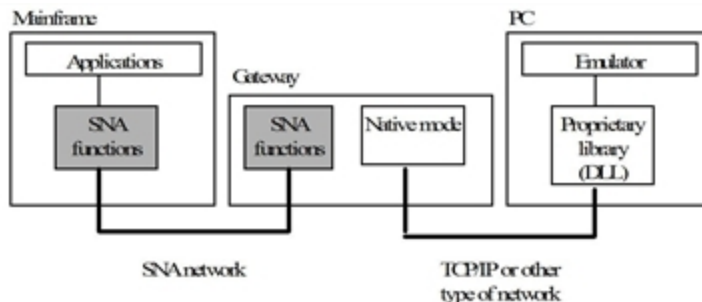
Microsoft SNA Server gateways

Esker's IBM 3270 emulator lets you connect to Microsoft SNA Server gateways in FMI or LUA native mode:

- FMI is the access mode used for access to Terminal (3270) or Printer (3287) type LUs defined on the gateway. These latter must correspond to a terminal or printer LU on the MainFrame server.
- LUA is the access mode used to access type 0, 1, 2 and 3 LUs on the MainFrame server, in other words terminals and printers, but also other types of LUs.

The emulator communicates with a DLL library that's specific to the gateway on the PC. The DLL then manages the connection with the gateway.

- In FMI native mode, the emulator communicates with library Snadmod.dll, supplied with the SNA client.
- In LUA native mode, the emulator communicates with library Winrui32.dll, supplied with the SNA client and which implements the RUI API allowing use of type 0, 1, 2 and 3 LUs.



SNA Server gateway prerequisites

To use Esker's 3270 IBM emulator in native mode with a Microsoft SNA Server gateway, you must first install the client part of SNA Server on the PC (SNA Server Client, version 3.0) supplied with Microsoft® BackOffice®. See Microsoft's documentation for installation details.

The connection parameters are grouped together in the 3270 emulator. Make sure the Microsoft SNA Server is correctly configured before connecting with the 3270 emulator.

5250 gateways

The gradual adaptation of SNA communication protocols (IBM) to other protocols like TCP/IP (Internet) has created a need for SNA gateways.

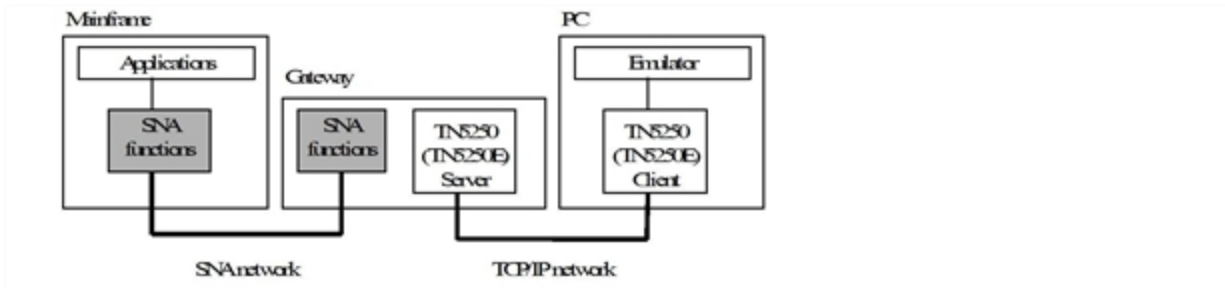
Esker's IBM 5250 emulator lets you connect to AS/400 systems directly via a TCP/IP network, using the TN5250 protocol, or via SNA gateways:

- UNIX SNA-TCP/IP gateways in TN5250 or TN5250E mode .
- Netware for SAA gateways via TCP/IP network or other, using standard commercial APPC APIs (Application Program-to-Program Communication) .
- Microsoft SNA Server gateways via TCP/IP network or other, thanks to standard commercial APPC APIs (Application Program-to-Program Communication) .

UNIX SNA-TCP/IP gateways

Esker's **5250** IBM emulator lets you connect to SNA-TCP/IP gateways on UNIX machines in TN5250 or TN5250E.

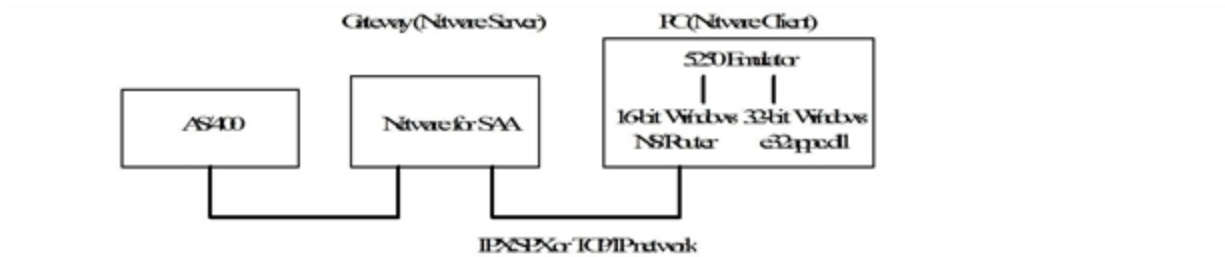
The PC emulates a 5250 terminal (or 5250E) by connecting to the gateway as if it was a server.



NetWare for SAA gateways

Netware for SAA is integrated as follows:

- The emulator sends its commands to an APPC function library named e32appc.dll, supplied by IBM.
- The API transforms these requests into statements that comply with the LU6.2 protocol.



Netware for SAA prerequisites

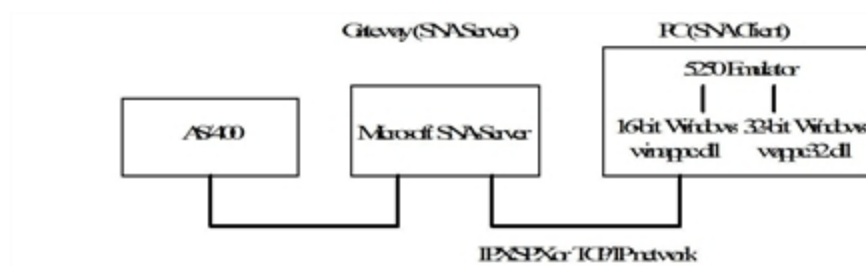
The PC running the emulator must have Novell's Netware client (which comprises API e32appc.dll) as well as Novell's Intranetware for SAA client:

The connection parameters are grouped together in the 5250 emulator. Make sure the Novell server is correctly configured before connecting with the 5250 emulator.

Microsoft SNA server gateway

Microsoft SNA server is integrated as follows:

- The emulator sends its commands to an APPC function library supplied by IBM (wappc32.dll).
- The API transforms these requests into statements that comply with the LU6.2 protocol.



Microsoft SNA Server gateway prerequisites

The PC running the emulator must have the SNA Client (SNA Server Client, version 3.0 supplied with Microsoft® BackOffice®), including the API wappc32.dll.

The connection parameters are grouped together in the 5250 emulator interface. Make sure the Microsoft SNA Server is correctly configured before connecting with the 5250 emulator.

Starting the synchronous emulator

Run the program by clicking the IBM Mainframe Access or AS/400 IBM Access icon from Application Access in the Esker Tun group.

Under Windows 8, right-click an empty area of the Start screen and click All apps. To start the application, find and click its tile.

User interface

Menus

Note:

The menus and options described below may not all be available if you are using a customized version of the emulator with some of the menus and options removed.

- File: Open, close and save a session, print and set up, printing, quit the emulator
- Edit: Copy/paste options, cut, select all the text

- Settings: Font, colors, hotspots, character tables, keyboard,, function-key panel, background settings, Print template definition, Emulation options, Association of a startup or end macro, Emulation customizer, Display (full screen, toolbar, status bar, OIA bar, function-key panel), trace or log file
- Transfer (only in 3270 emulation): File transfer configuration, receive and send files, multiple file transfer
- Macro: execution, edit, save, encrypt string,
- Associated printer: association of a printer for 3287 or 3812 printing
- Options: firewall implementation (proxy) , start other Tun suite tools, change language
- Help: Online help and About dialog box.

Menu display

Whatever menus are available in the emulator, you can show or hide the menu bar. By hiding the menu bar, you can display the emulator in full screen mode like a dumb terminal.

To display the emulator in full screen, choose Full Screen from the Settings menu. If you have already hidden the menu bar, right-click anywhere in the screen.

If you want to change the emulator display the next time it's used, choose Options from the Settings menu, and then select the Full Screen Display option in the displayed dialog box.

Note:

If you are an integrator of emulation solutions, you can prepare customized versions of the program for your users. You can select the menus and options to include in the customized version. For more details, see “Emulation customizer” on page 98.

Toolbar


The toolbar lets you access the emulator's main commands simply and quickly.























With all the versions of the emulator, you can show or hide the toolbar (Toolbar option in the Settings menu).

You can customize the toolbar:

- Floating toolbars in the emulation screen, docking at the sides of the screen: To do that, click the toolbar gripper and hold the left mouse button down, and move the toolbar. When the pointer reaches one of the sides, the toolbar docks automatically .
- Toolbar style: You can choose flat (default) or not (choose Options from the Settings menu and select the Flat toolbar style option in the Session tab).

The following buttons are available:

| | |
|---|--|
|  | Opens a new emulation session and closes the current session if there is one |
|---|--|

-
-  Loads all the settings used in a session (host address, colors, character set, keyboard, etc.) from a setup file with a .cfs extension
 -  Saves all the settings used in a session (host address, colors, character set, keyboard, etc.) in a single file with a .cfs extension
 -  Prints screen page(s) with template
 -  Prints the screen or a selected area of the screen in text mode
 -  Prints the screen or a selected area of the screen in graphic mode
 -  Sends a file to the server (only in 3270 emulation)
 -  Receives a file from the server (only in 3270 emulation)
 -  Cuts the selection highlighted with the mouse and copies the contents to the clipboard
 -  Copies the selection (selected with the mouse) to the clipboard
 -  Copies the contents of the clipboard to the cursor position
 -  Deletes the selection
 -  Selects the screen display font
 -  Selects the next font size up to the current one. To use this option, the current font must exist in different sizes. This is the case with the SystemPC font supplied with the product.
 -  Selects the next font size down to the current one. To use this option, the current font must exist in different sizes. This is the case with the SystemPC font supplied with the product.
 -  Associates a color to the emulation session's monochrome attributes.
 -  Configures hotspots.
 -  Allows the modification of the correspondence table for the PC and extended EBCDIC character sets
 -  Allows you to change the values returned by the keyboard
 -  Allows you to print with a template
 -  Changes the emulator options
 -  Shows or hides the function-key panel
 -  Executes a macro
-

Status bar / OIA bar

Choose Settings > Status Bar or Settings > OIA Bar to toggle these bars on and off.

These bars show information about the current session (data expected, keyboard locks, etc.) and cursor coordinates. The OIA bar is specific to the IBM environment.

Connection in IBM synchronous emulation

Connection in IBM 3270 synchronous emulation

Esker's IBM 3270 emulator enables you to make the following types of connection:

- Direct connection to a MainFrame server or connection through a UNIX SNA-TCP/IP gateway using the TN3270 or TN3270E protocols.
- Connection to a MainFrame server via a Microsoft SNA Server or Novell Netware for SAA gateway.

Connection in IBM 5250 synchronous emulation

The Esker IBM 5250 emulator enables you to make the following types of connections:

- Direct connection to an AS/400 system or connection through a UNIX SNA-TCP/IP gateway using the TN5250 or TN5250E protocol.
- Connection to an AS/400 system via a Microsoft SNA Server or Novell Netware for SAA gateway.

Multiple connection in IBM 3270 or 5250 synchronous emulation

The Esker IBM 3270 and IBM 5250 emulator propose a type of connection called multiple connection.

This type of connection enables you to define several configurations for connection to a MainFrame server or a given AS/400 system. If the first connection configuration used fails, the next configuration is used. The order in which the configurations are used, can be:

- The order given by the list of configurations, or
- A random order.

This type of connection is particularly interesting if you have a number of redundant gateways for accessing a given server: it enables you to connect to this server without stipulating the gateway to be used. If a gateway is momentarily unserviceable, the connection will be via another one in a way imperceptible to the user.

Customizing a synchronous emulation session

To open an emulation session, choose New Session in the File menu.

3270 emulation, TN3270E connection

This box contains two tabs:

- The Session tab enables you to specify the type and the parameters of the connection to be made.

- The Status tab gives the state of the connection (the state is Not Connected at the moment of connection).

Options common to the Session tab

Selecting the Close Session on Confirmation check box displays a confirmation dialog box on exiting the program.

If the Startup Connection check box is selected, the program will automatically reconnect at the host.

If the Reconnection after a shutdown check box is selected, the program will automatically re-establish the connection if it's interrupted by the host.

Session tab in TN3270 or TN3270E connection (3270 emulation)

This section describes the Session tab as it appears in 3270 emulation when the connection type is TN3270 / TN3270E.

Terminal Type

This field sets the type of 3270 terminal display you want to use. There are four possible models:

- Model 2 (24 x 80)
- Model 3 (32 x 80)
- Model 4 (43 x 80)
- Model 5 (27 x 132)

The Detects the model check-box enables you to change the terminal model specified at connection in order to meet the needs of the application used. Select this check-box if you want to adapt the terminal model automatically to the application used.

Host name

Enter the name of the server or its IP address. You can also select a server from the drop-down list (only enter a name if you have a names server). In the case of a connection via a gateway, the host name corresponds to the gateway name.

Port

The default port number is 23. You can change this number, if necessary.

Protocol extension

The check-box Use TN3270E is selected by default: the TN3270E protocol is used to make the connection and if the server contacted cannot support the TN3270E, a TN3270 connection is then negotiated.

Deselect this check-box if you want to use the TN3270 protocol exclusively, without extension.

Logical unit name (only for TN3270E connection)

Enter the name of the logical unit (LU) you want to work on. The LU gives the type of resource to which you are connecting on the MainFrame server.

Session tab in SNA Server or Netware for SAA connection (3270 emulation)

This section describes the Session tab as it appears in 3270 emulation when the connection type is SNA Server LUA, SNA Server FMI or Netware for SAA.

Terminal Type

This field sets the type of 3270 terminal display you want to use. There are four possible models:

- Model 2 (24 x 80)
- Model 3 (32 x 80)
- Model 4 (43 x 80)
- Model 5 (27 x 132)

The Detects the model check-box enables you to change the terminal model specified at connection in order to meet the needs of the application used. Select this check-box if you want to adapt the terminal model automatically to the application used.

Logical unit name (only for TN3270E connection)

Enter the name of the logical unit (LU) you want to work on. The LU gives the type of resource to which you are connecting on the MainFrame server.

TN5250 or TN5250E Session connecting tab (emulation 5250)

This section describes the Session tab as it appears in 5250 emulation when the connection type is TN5250/TN5250E.

Terminal type

Lets you choose the type of 5250 terminal.

All the terminals listed can handle 24 lines and 80 columns, but only three of them, IBM-3180-2, IBM-3477-FC and IBM-3477-FG, can display 27 lines with 132 columns:

- IBM-3179-2
- IBM-3180-2
- IBM-3196-A1
- IBM-3477-FC
- IBM-3477-FG
- IBM-5251-11

- IBM-5291-1
- IBM-5292-2

Host name

Enter the name or IP address of the server or select it from the drop-down list (only enter a name if you have a names server). In the case of a connection via a gateway, the host name corresponds to the gateway name.

Port

The default port number is 23. You can change this, if necessary.

Protocol extension

The check-box Uses TN5250E is selected by default: the TN5250E protocol is used to make the connection and if the server contacted cannot support the TN5250E, a TN5250 connection is then negotiated.

Deselect this check-box if you want to use the TN5250 protocol exclusively, without extension.

Unit name

Optional field. If you enter nothing in the field, the AS/400 system will attribute a default name to your client PC to uniquely identify it. You can, however, enter a terminal name to identify your machine.

Click on the Advanced button to customize the advanced properties of the TN5250E session.


Identification


To enable automatic login, fill the required parameters when login to the AS/400 system: user name and password, library, menu and program (these settings correspond to the fields that appear on the login screen of an AS/400 system).

Select the Encrypt check box if you want the identification password to be crypted on login.

Terminal parameters

You can modify the default configuration of the AS/400 terminal on which the connection is established. Specify a new Code Page and/or Charset to use if necessary. Click Default to reset these values to your session defaults (given by the character set).

The UserVar Text and UserVar Binary lists let you configure other advanced parameters of the terminal. To add an additional IBM parameter (text or binary), click . Then, enter the name of the IBM parameter in the upper left part of the field, and its associated value in the right part.

To remove a character from the list, select it and click .

Refer to your IBM documentation for more details on these settings.

Session tab (Netware for SAA or SNA Server connection on 5250 emulation)

This section describes the Session tab for 5250 emulation when the connection type is Netware for SAA or SNA Server.

Terminal type

Lets you choose the type of 5250 terminal.

All the terminals listed can handle 24 lines and 80 columns, but only three of them, IBM-3180-2, IBM-3477-FC and IBM-3477-FG, can display 27 lines with 132 columns:

- IBM-3179-2
- IBM-3180-2
- IBM-3196-A1
- IBM-3196-A1
- IBM-3477-FC
- IBM-3477-FG
- IBM-5251-11
- IBM-5291-1
- IBM-5292-2

System name

Enter the name or IP address of the server or select one from the drop-down list (only enter a name if you have a names server).

Unit name

Optional field. If you enter nothing in the field, the AS/400 system will attribute a default name to your client PC to uniquely identify it. You can, however, enter a terminal name to identify your machine.

Click on the Advanced button to customize the advanced properties of the 5250 session via a Netware for SAA or SNA Server gateway:

Mode name

By default, the mode name is QPCSUPP. This mode is specific to the gateway and APPC library. Typically, you don't need to change this value. Contact your network administrator if you want to change it.

Local LU name (only for SNA Server connections)

For connections through an SNA Server gateway, enter the name of the local LU as it's registered on the gateway. This field must be completed.

This field isn't shown for Netware for SAA gateway connections. The name of the local logical unit must be correctly registered on the gateway for the connection to succeed.

User name/Password


Enter the user name and associated password that allow you to connect to the AS/400 system. The user name and password are defined on the gateway.

Multiple connection Session tab

This section describes the Session tab in 3270 or 5250 emulation, when the type of connection selected is Multiple connection.

Connection configurations


The list gives all the connection configurations already defined.

To define a new connection configuration to add to the list, click on the  button. Then fill out the connection parameters as if you were defining a new connection.



To change the parameters of an existing connection configuration, click on this configuration.

Note:

For more details on the parameters of a connection configuration, refer to the description of the Parameters tab corresponding to the type of connection for the configuration you want to add or modify.

To delete a connection configuration from the list, select it and click on the  button.

Connection configuration order

To use the configurations in the order given by the list, deselect the Random selection check-box. The configurations will be tried in this order, one after the other, until the connection is successful or until all the configurations have been tried.. Then click on the  and  buttons to change the position of the connection configurations in the list, and define the order in which they are to be used.

To use the configurations in a random order, select the Random selection check-box. The configurations will be tested in a random order, until the connection is successful or until all the configurations have been tried. This method is useful for avoiding systematic overload of the servers used in the connection configurations at the beginning of the list.

Wait time before using the next connection configuration

The Timeout field enables you to specify the maximum wait time (in seconds) after which a connection attempt is considered to have failed: the next connection configuration is then used (if there is one).

Note:

This delay is identical for each connection attempt.

Starting sessions

After choosing a host and a terminal, click OK to connect. If TCP/IP is running correctly on the PC and the host, you'll receive the sign-on screen.

The OIA bar

The OIA bar (Operator Information Area) is an area which appears at the bottom of the screen in an IBM emulation session. It gives information on the status of the connection to the host machine, taking the form of messages or symbols.

> Display/Hide OIA bar

Select or deselect the OIA Bar option from the Settings menu.

Content of OIA bar

The OIA bar uses columns 1 to 80 on the emulation screen (including for a 132-column terminal model).

It contains the following information:



- Columns 1 to 3: emulation session indicators.
- Columns 9 to 17: keyboard locking indicators.
- Columns 26 to 27: message reception indicator (5250 only).
- Columns 32 to 41: APL mode indicator.
- Columns 42 to 44: Shift mode indicator.
- Column 52: insert mode indicator.
- Columns 75 to 80: indicator of cursor position on screen.

Emulation session indicators (Columns 1 to 3)

These indicators appear when the PC is connected to a host machine.

Controller type (Column 1)

A character in Column 1 means that the session is using a controller. The character used specifies the type of link to this controller.

The following characters or symbols can appear:

-
- DFT connection to the host machine via a 3274 controller.
 - S DFT connection to the host machine via a 3174 controller.
 - N DFT connection to the 9370 host machine.
 - V Connection to the host machine via APPC.
 - M Other type of connection to the host machine.
-

Connection type (Column 2)

-
- A The connection method is not SNA (connection type TN3270 , TN3270E or TN5250).
 - B The connection method is SNA (connection via an SNA Server or Netware for SAA gateway).
-

Host machine access type (Column 3)

The PC is connected to the host machine via an unknown session type.

A symbol indicates the type of access made by your PC to the host machine:

-
- The PC is connected (or ready to connect) to an application on the host machine via a LU-LU type session.
 - ⊠ The PC is connected to the host machine via a LU-SSCP type session.
 - ⊛ The PC is connected to the host machine via an unknown session type.
-






Keyboard locking indicators (Columns 9 to 17)

These indicators appear when the PC's keyboard is locked by the host machine. All these indicators contain an «X» in column 9, which means that you should enter nothing on the keyboard. This X is combined with other symbols in columns 11 to 17, stating why the keyboard is locked.

The Reset function is used to unlock the keyboard. The key corresponding to the Reset function is defined by the type of PC keyboard used (generally the Ctrl key on the PC keyboard).

The following characters or symbols can appear:

-
- X ⊚ The host machine needs time to run a function.
 - X SYSTEM The host machine has locked your keyboard. Wait until a message appears, or press Reset.
-

| | |
|---|---|
| X ←  → | An operation has been carried out in an incorrect screen zone. Press Reset and repeat the operation, positioning the cursors in the correct zone. |
| X  >  | The keyboard buffer is activated and full. Press Reset. |
| X  NUM | Non-numerical data has been entered in a screen zone reserved for numerical data. Press Reset and repeat the operation, positioning the cursor in the correct zone. |
| X  > | Too much data has been entered in a screen zone. Press Reset. |
| X PROGnnn | The emulator has detected a program error in the data received by the host machine. Press Reset and repeat the operation (if possible). |


Message indicator (5250 only) (Columns 26 -27)

MW: A message has been received and placed on hold.

APL mode indicator (Columns 32 to 41)

APL: Activation of APL mode on PC keyboard.

Shift mode indicator (Columns 42 to 44)

Ò: Activation of the Shift key () on the PC keyboard.

ALT: Activation of the Alternate Shift key (Alt Gr) on the PC keyboard.

Insert mode indicator (Column 52)

^: Activation of PC keyboard insert mode (Ins key).

Cursor position indicator (Columns 75 to 80)

XX/YYY: Row/Column position of the cursor on the emulation screen; XX = row number, YYY= column number.

Program Check Messages

PROG 705: An SNA message was received out of sequence.

PROG 706: An SNA message was received with incorrect chaining.

PROG 707: An SNA message was received with incorrect bracketing.

PROG 708: An SNA message was received before data traffic was started.

PROG 709: An SNA message was received that violated half-duplex protocol.

PROG 711: An SNA Bind command was received when the logical terminal was already bound.

PROG 714: An SNA message was received before the session was bound.

PROG 715: An SNA message was received before the session was activated.

PROG 717: An SNA Bind command that specified incorrect FM or TS profiles was received.

PROG 718: An SNA Bind command that specified an incorrect primary protocol was received.

PROG 719: An SNA Bind command that specified an incorrect secondary protocol was received.

PROG 720: An SNA Bind command that specified an incorrect common protocol was received.

PROG 723: An SNA Bind command that specified an incorrect LU type was received.

PROG 729: A Read Partition command was received while the workstation was in retry state.

PROG 750: An invalid 3270 command was received.

PROG 751: An order with an invalid charset ID was received.

PROG 752: A Set Buffer Address, Repeat to address, Erase Unprotected To Address or Modify Field order that specified an invalid address was received.

PROG 753: A Graphic Escape order that specified an invalid character code was received. Or a Start Field Extended, Modify Field, Set Attribute, Set Reply Mode order that specified an invalid attribute value was received.

PROG 754 : One of the following commands was received without required parameters: Set Buffer Address, Repeat To Address, Erase Unprotected To Address, Start Field, Start Field Extended, Modify Field, Set Attribute, Graphic Escape.

PROG 755: Invalid character code was received.

PROG 755: A Write Structured Field command was received with an invalid structured field.

PROG 758: A Set Reply Mode command was received with an invalid structured field length.

PROG 759: A Write Structured Field command was received with an invalid structured field length.

PROG 760: A Write Structured Field command was received with reserved fields not zero.

PROG 761: A Write Structured Field command was received with invalid partition identifier.

PROG 771: A Write Structured Field command was received with an invalid structure field command.

APL Mode(3270 emulation)

APL characters are semi-graphical characters that are used on MainFrame servers, to draw up frames for instance. Esker's 3270 emulator allows you to use this type of characters in your emulation sessions. To use APL characters, you only need to switch from the normal mode to the APL mode.

Switching to the APL Mode

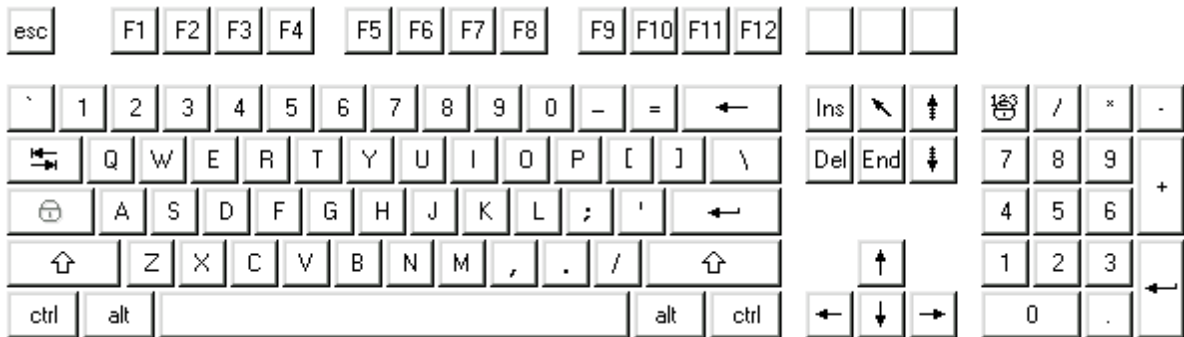
Use the Alt Gr+F8 (or Ctrl+Alt+F8 if your keyboard doesn't have the Alt Gr key) default keyboard key combination. The APL message appears in the OIA bar.

When in APL mode, use the Alt Gr+F8 (or Ctrl+Alt+F8) key combination to switch back to normal mode.

Note:

The PC keyboard cannot be edited in the APL mode.

The values returned by the PC keyboard keys in APL mode are displayed below, depending on the combination used (none, combination with the Shift key or combination with the Alt key). The PC keyboard used for this description is of the following type:

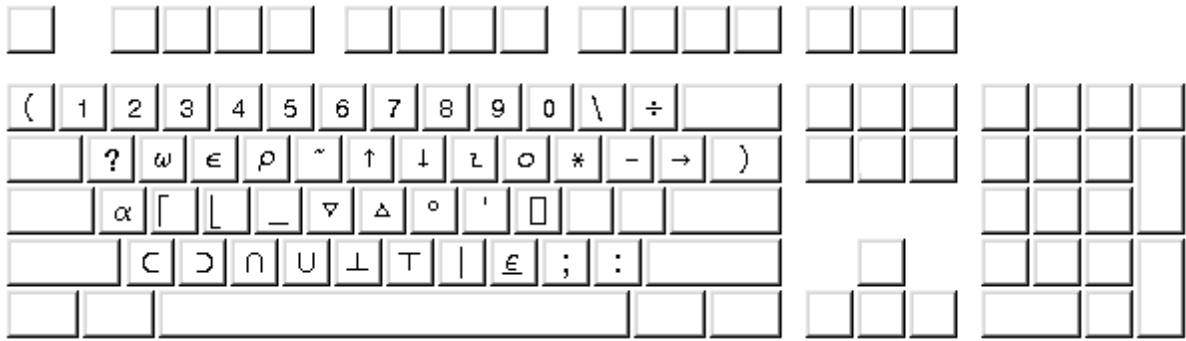


Depending on the language and type of your PC keyboard, you might need to adjust the position of specific keys.

APL keyboard (without key combination):



APL keyboard combined with the Shift key:



APL keyboard combined with the Alt key:



HLLAPI (3270 emulation only)

HLLAPI (High Level Language Application Interface) is a set of functions that makes it possible to program in high level languages such as C, Pascal, Basic and even COBOL.

With HLLAPI functions, the programs you write behave in the same way, as far as the host is concerned, as the user of a 3270 terminal. The HLLAPI interface identifies the functions and the data structures that are used, and carries out the operations defined in the user's program on the remote machine.

You can use the HLLAPI interface with Esker's 3270 synchronous emulator. That way, you can transparently access data on a mainframe server from a 3270 emulation session using Windows applications you've written in C, C++ or Visual Basic.

Esker's synchronous emulator uses an HLLAPI interface that's compatible with IBM's EHLLAPI and Microsoft's WinHLLAPI: All the functions defined by these interfaces are supported. The HLLAPI functions used by the Windows applications you write are contained in the file whll32.dll supplied with Tun PLUS.



Using HLLAPI

To use the HLLAPI interface, you must use HLLAPI functions when you program your applications and install the function library whll32.dll on your PC.

A Windows HLLAPI application can call one or more emulation sessions. To identify the different emulation sessions, the program uses the short name attributed to each session. Since the short name is a letter of the alphabet, the maximum number of simultaneous 3270 sessions that can be opened is limited to 26.

To define a short name for a 3270 emulation session, do the following:

Click the TN3270 icon in the Tun EMUL group to start Tun 3270.

Under Windows 8, right-click an empty area of the Start screen and click All apps. To start the application, find and click its tile.

Choose File > New Session... from the main menu to open a new emulation session.

On the HLLAPI tab, click the letter of the alphabet you want to use as a short name.

Note:

The short names currently used by other 3270 emulation sessions running on your PC appear grayed.

You can enter a comment describing your session in the Long Name field (optional) on this tab.

You can change the short and long names once the session is started by clicking the emulation screen with the right mouse button and choosing Session Properties... from the context menu.

Windows can use all the HLLAPI functions implemented and exchange data with the opened emulation session. This mechanism makes it possible to access 3270 emulation data easily from a user-friendly Windows interface.

For more information, refer to one of the HLLAPI norms describing the available functions and the way to implement them in your programs.

Refer to the example “Accessing data on an IBM MainFrame server from a Windows application (HLLAPI)” below, to better understand how to use HLLAPI with Tun.

Accessing data on an IBM MainFrame server from a Windows application (HLLAPI)

The following HLLAPI example shows how to access the data in a 3270 emulation session from a Windows application without displaying the 3270 emulator screen at all. Using this principle, you can completely change an emulation session's interface, making it more user-friendly while maintaining full 3270 functionality.

The application in the example uses four main functions:


Connect: Establishes a connection between the emulation session and the application.


Disconnect: Breaks the connection with the emulation session.


GetScreen: Copies the contents of the emulation screen to the application.


SendString: Sends a string of characters to the emulation session screen.

To proceed, Tun 3270 must be running with an opened session with the short name A. You can then perform the following operations:

Click the toolbar icon  to establish the connection between the application and the 3270 emulation session with the short name A.

Click the toolbar icon  to send a character string to the emulation session, just as if you entered the string directly in the 3270 emulation screen.

Click the toolbar icon  to copy the contents of the emulation session screen to the application window. You can then use this data.

Click the toolbar icon  to disconnect the application from the emulation session.

Programming

The five steps described below refer to the code source in the sample application shown at the end of this chapter. The steps show how to declare the whll32.dll DLL (step 1) and use its functions (steps two through five).

Note:

The sample source code (in Visual Basic 5.0) was copied to the Tun EMUL installation directory when Tun EMUL was installed (C:\... \Tun\Emul\Demo\Hllapi). The code is in the files frmMain.frm, frmSend.frm, frmAbout.frm, Module1.bas, HLLAPI.vbp.

STEP 1: Declare functions that call the DLL whll32.dll

The DLL routines are stored in files that are external to the Visual Basic application files. You must, therefore, indicate the location of these routines in your application and the arguments they take. You must supply this information for each routine by declaring the function in the Declarations section of the Visual Basic module. Once you've declared the DLL routine, it can be called by any Visual Basic function using a Call statement.

You must declare the following two DLL routines from whll32.dll:

WinHLLAPIStartup(): Enables the application to specify the required version of Windows HLLAPI and other information on its Windows HLLAPI implementation. This function must be called before using Windows HLLAPI functions. It returns a value indicating if the given version is supported and if the declaration of its HLLAPI implementation was properly made.

WinHLLAPI(): Enables the application to call functions from the DLL, specifying the necessary parameters (they aren't always all used and their type depends on the function).

HLLAPI syntax:

```
int WinHLLAPIStartup(WORD wVersionRequired, LPWHELLAPIDATA lpData)
```

wVersionRequired: Windows HLLAPI to support

lpData: Structure containing information on the HLLAPI implementation

```
extern VOID FAR PASCAL WinHLLAPI(LPWORD lpwFunction, LPBYTE lpbyString, LPWORD lpwLength, LPWORD lpwReturnCode)
```

lpwFunction: Number of the function to call

lpbyString: String used to pass data (from the application to WinHLLAPI or vice-versa)

lpwLength: Length of the string passed

lpwReturnCode: Return code showing status of the function called

STEP 2: Defining the connection procedure

This procedure first starts the the WinHLLAPIStartup function, then calls the HLLAPI function number 1: Connect Presentation Space. This function establishes a connection between a given session on the host machine (presentation space) and the Windows HLLAPI application. The session the connection is established with, is defined by its "short name" passed as a parameter. In the example code, the short name passed as a parameter is A: The connection will be established with the emulation session that is defined by the short name A.

The syntax for this function is as follows:

```
WinHLLAPI(CONNECTPS, lpbyString, lpwLength, lpwReturnnnCode)
```

```
CONNECTPS (lpwFunction): 1
```

lpbyString: Alphabetic character identifying the session (short name) when the function is called

lpwLength: Not used (1 by default)

lpwReturnnnCode: Return code indicating connection status

STEP 3: Defining the disconnection procedure

This procedure calls HLLAPI function number 2: Disconnect Presentation Space. The function disconnects a session with the host machine.

The syntax for this function is as follows:

```
WinHLLAPI(DISCONNECTPS, lpbyString, lpwLength, lpwReturnnnCode)
```

```
DISCONNECTPS (lpwFunction): 2
lpbyString: Not used
lpwLength: Not used
lpwReturnnnCode: Return code indicating connection status
```

STEP 4: Defining the procedure for retrieving the emulation session screen

This procedure calls HLLAPI function number 5: Copy Presentation Space. This function copies the emulation session's current screen contents to a character string buffer. In the example, this character string is copied to the variable screen which is then displayed in the application's window.

The syntax for this function is as follows:

```
WinHLLAPI (COPYPS, lpbyString, lpwLength, lpwReturnnnCode)
COPYPS (lpwFunction): 5
lpbyString: Character string the emulation session's screen's contents are copied to (when you define this variable, you must assign it at least the number of bytes required by the maximum screen size)
lpwLength: Not used
lpwReturnnnCode: Return code indicating the status of the screen copy
```

STEP 5: Defining the procedure for sending data to the emulation session

This procedure calls HLLAPI function number 3: Send Key. Function number 5 sends a key sequence (the maximum number is 255) to the connected emulation session. These key sequences are displayed in a session as if they'd been entered by a user. In the example, the string StringToSend that was entered by the user in an application window, is sent.

The syntax for this function is as follows:

```
WinHLLAPI (SENDKEY, lpbyString, lpwLength, lpwReturnnnCode)
SENDKEY (lpwFunction): 3
lpbyString: String containing the sequence of keys sent when the function is called
lpwLength: Length of this strength in bytes
lpwReturnnnCode: Return code indicating the status of the string send
```

Extract from the corresponding Visual Basic code (Declarations part of the module):

```
Type WHLLAPIDATA
    wVersion As Integer
    szDescription(128) As Byte
End Type
Public Declare Function WinHLLAPIStartup Lib "WHLL32.DLL" (ByVal Version As Integer, ByRef lpData As WHLLAPIDATA) As Integer
Public Declare Sub WinHLLAPI Lib "WHLL32.DLL" (ByRef lpwFunction As Integer, ByVal lpbyString As String, ByRef lpwLength As Integer, ByRef lpwReturnCode As Integer)
```

```
Public fMainForm As frmMain

Sub Main()
    Set fMainForm = New frmMain
    Load fMainForm
End Sub

Sub Connect()
    Dim dat As WHLLAPIDATA
    code = WinHLLAPIStartup(1, dat)
    Call WinHLLAPI(1, "a", 1, code)
End Sub

Sub Disconnect()
    Dim dat As WHLLAPIDATA
    Call WinHLLAPI(2, "", 0, code)
End Sub

Sub GetScreen()
    Dim screen As String * 2000
    Call WinHLLAPI(5, screen, 0, code)
    fMainForm.Text1.Text = screen
End Sub

Sub SendString(StringToSend As String)
    Call WinHLLAPI(3, StringToSend, Len(StringToSend), code)
End Sub
```

Customizing the emulator

Depending on the needs of your type of emulation, you can configure the number of lines and columns in your application, add scroll bars, choose the type of line scrolling and cursor coupling, and center the terminal in the emulator window or not. You can also have a bitmap image of your emulation session as the screen background.

Asynchronous emulator

Choose Display Settings in the Settings menu.

The Screen tab concerns configuration of the emulation screen (dimensions, scrolling, etc.), and the Background concerns the display of the bitmap image on the screen background.

Screen size

The default setting for Esker's asynchronous emulator emulates 80 x 25 screens. You can change this setting in the Lines Used and Columns Used fields. This is useful for emulating terminals that use different dimensions (for example, 132 columns or 43 lines).

Typically, only 25 lines are simultaneously stored by the asynchronous emulator. You can change this value in the Lines Stored field. The maximum value for this field is 1024. Use the scroll bars on the right of the emulation screen to view the lines stored by the asynchronous emulator that haven't been displayed yet.

Scroll type

You can choose a type of scrolling if you want to display the scrollbars:

- Jump: enter the number of lines scrolled per jump.
- Smooth: adjust the scroll speed as you wish.

Scroll bars

The Vertical and Horizontal check boxes toggle the display of horizontal and vertical scroll bars along the side and bottom of the emulation screen. The scroll bars are useful if you've opted for more than 25-line storage and you're not using Dynamic Sizing.

> For more information about Dynamic Sizing, see "Dynamic Sizing" on page 66.

Cursor coupling

The Horizontal and Vertical check boxes let you enable or disable cursor coupling in either direction. Cursor coupling scrolls the screen so that the cursor is always visible. If

either check box is cleared you can move the cursor outside the range of the terminal window.

Terminal centering

Select the Center terminal check box to center the terminal window into the emulation window. While the terminal screen is centered in the emulation window, you can put a frame round it. To do that, choose the Frame terminal check box.

Note:

The fields described below are those of the Background tab.

File

This field should hold the name of the bitmap file (.bmp) with the required image. By default the program looks for the .bmp extension files in the asynchronous emulator's installation directory.

Alignment

The vertical and horizontal alignment options give the position of the image in the emulation window. These options are only of use when the image is not adjusted to fit the display zone.

Tile

If the image is smaller than the emulation screen, it can be duplicated until it fits all the available space by selecting this check box.

Scroll with text

If the bitmap is tiled and the display zone is not the window, you can scroll the bitmap with the text if you select this check box.

Fit to display zone

If the bitmap doesn't coincide exactly with the size of the emulation screen, you can resize it until it occupies all the available space by selecting this check box.

Display zone

The image display zone can be:

- Full memory: the image location is relative to the zone bounded by all the lines memorized.
- Terminal memory: the image is displayed in the zone occupied by the emulated terminal lines.
- Window: the image location is relative to the window, independently of the number of lines displayed or memorized.

Synchronous emulator

Choose Background in the Settings menu.

Image

Use the Browse button to look for the bitmap image you want to display as the screen background, or enter the file's absolute path.

Select the display options:

- None: no image displayed. The background stays as-is.
- Tile: the image is tiled over the background.
- Center: the image is centered in the middle of the screen.
- Adjust to window: the image is adjusted to fill the entire screen.

Display frame

You can add a frame around the terminal by selecting the Display a frame around terminal check box.

Choice of font

You can choose a font different from that displayed on the screen by default. You can also alter the size.

Any non-proportional character font available under Windows can be used in emulation. To use all the semi-graphic characters correctly, the font needs to be OEM, and not ANSI.

The SystemPC font is supplied by Esker for the best possible use of emulation with Windows. It's a fixed OEM font ranging in size from 4 to 30 that was developed especially for the asynchronous emulator.

Note:

Use Sys132PC for emulation with 132 columns.

Asynchronous emulation

Choose the Display Settings option in the Settings menu, and then click the Font tab.

Font used

You can define the type and size of font that's used by the emulator in these modes (80 or 132 columns). The 80 columns and 132 columns options let you select the character font, irrespective of the width of the emulated terminal.

Dynamic Sizing

This function causes the emulator to change the size of the font in keeping with the size of the emulation window so that a full terminal screen is always displayed (80 x 25 or 132 x

25, depending on the terminal). Dynamic sizing is best used with character fonts that are available in multiple sizes (as is the case with SystemPC or Sys132PC).

Synchronous emulation

Choose the Font option from the Settings menu.

Font used

You can define the type and size of font that's used by the emulator.

Dynamic Sizing

This function causes the emulator to change the size of the font in keeping with the size of the emulation window so that a full terminal screen is always displayed. Dynamic sizing is best used with character fonts that are available in multiple sizes (as is the case with SystemPC).

Customizing colors

By default, the characters appear on your PC screen as they are displayed on the terminal. You can however change the various attributes displayed (Normal attribute, Reverse attribute, Underline attribute, etc.) by giving them colors and styles of your choice (bold, italic, underline).

Asynchronous emulation

Choose the Display Settings option from the Settings menu, and then click the Colors tab.

Note:

You can access this configuration box directly from the screen with the mouse: display the context menu by clicking on the character whose color you want to change, and select the Edit colors option.

This color configuration box enables you to customize the character display on the screen depending on the type of attributes or the color of the character or the background in the application.

For each attribute or color, you can choose one color for the text and another for the background.

Selection of attributes

To modify the attributes, select the Attributes option.

The character attributes generally supported by traditional applications are:

- Normal
- Blink
- Dim
- Highlight

- Reverse
- Underline
- Other font
- Protected

You can combine several attributes.

By default, the Normal attribute is selected (no box is checked). To redefine another attribute, select the corresponding check box. To redefine a combination of attributes, select the corresponding check boxes.

Capture the attribute with the mouse

You can select an attribute directly by capturing it on the screen. To do this, click on the



button in the Attributes section. The color configuration box disappears: use the mouse cross-hair cursor to click on the character for which you want to redefine the attribute.

The color configuration box then reappears: the attribute(s) corresponding to the captured character are selected.

Note:

If you opened the configuration box from the context menu of a character displayed on the screen (Edit colors option), the attribute of the character concerned is selected when the configuration box is opened.

Color selection

Some applications return characters in a particular color or on a particular background. To redefine these characters, select the Colors option then choose the character or background color desired for the application from the drop-down list that appears.

Changing colors and styles

After selecting an attribute, a combination of attributes, or a character or background color, you can then apply the colors and styles you wish to this element.

With the Colors sub-tab, select the foreground and background colors you want. .

Depending on the type of attribute, the check box proposed is:

- Pre-defined colors: select this check box if you want to take pre-defined colors for the attribute selected (option for all the attributes except Reverse video and Underline).
- Base color effect: the Reverse video and Underline attributes can be automatically deduced from the Normal attribute. Select this check box to make this happen, or deselect it if you want to customize these attributes differently.

Attribute combinations that aren't associated with particular colors adopt the color of a sub-combination (for example, if the combination Blink + Dim + Reverse isn't associated with a color, it uses the color associated with Blink + Dim).

In the Styles sub-tab, select the text style (Italic, Bold, Underline) and the background character style.

Note:

To use the relief or shadow effects, it's better to choose a gray background.

All the options can be combined to produce the desired effect (Example: Bold + Inset + Outdent background).

Synchronous emulation

Choose the Colors option from the Settings menu.


This color configuration box enables you to customize your character display on the screen according to the type of attributes or, for 3270 emulation, according to the color of the character sent by the server.

For each attribute or color, you can choose one color for the text and another for the background.

Selection of attributes

Select the 3270 standard attribute or color you want to obtain from the drop-down list(s).

Attribute capture with the mouse

You can select an attribute directly by capturing it on the screen. To do this, click on the  of the Attributes section. The color configuration box disappears: with the mouse cross-hair cursor, click on a character for which you want to redefine the attribute.

The color configuration box then reappears: the attribute(s) corresponding to the captured character are selected.

After selecting an attribute or a color in 3270 emulation, you can then apply the colors and styles you wish to this element.

Altering the color

In the Colors sub-tab, select the foreground and background colors you want.

For 3270 emulation, click the Uniform Background Color button to assign the selected color to all the attributes.

For 5250 emulation, click the Apply Normal Attribute Background Color to All Attributes button so that all the attributes have the same background color as the normal attribute.

Display Attribute characters

In a string of characters there are special characters, represented on the screen by a blank, that contain information on the characters that follow (for example, on the format of a character string). Normally, these characters are the same color as the background and therefore don't appear on the screen.

However, in both 3270 and 5250 emulation these characters can be made to appear in a different color to make them visible. To do this, assign the color of your choice to the attribute Attribute Characters (at the end of the Attribute list), and select the Display Attribute Characters option to apply that color to these characters.

Style change

In the Styles sub-tab, select the font style (Italic, Bold, Underlined, Blink) and the style of the characters and borders.

Note:

To use relief or shadow effects, it's better to choose a gray background.

Working with a personal function-key panel

In the emulation window, you can display a key panel enabling you to access the main functions of your terminal with a simple click of the mouse: send a character string, function keys, execute actions via a script.

> **Show or hide function-key panel associated with the session**

Select or deselect the Key panel... option at the end of the Settings menu.

> **Start the function-key panel editor to create or modify a key panel**

Select the Function-Key panel... option at the beginning of the Settings menu.

> See "Function-Key Panel Editor" on page 123.

Asynchronous emulation

Choose the Display Settings option from the Settings menu, and then click the Function-key panel tab.

File

Select the function-key panel containing the function-key panel definition. Click on the Browse... button if the function-key panel does not appear in the default list proposed and look for the function-key panel of your choice.

Note:

The function-key panel files are text files with extension .pan.

Docking

The function-key panel can be transformed into a toolbar. To do this, select the docking option you want to apply to the function-key panel. The possible docking options are as follows:

- **Default:** The type of docking used is that defined when the function-key panel was created, in the panel parameters.
- > See the chapter “Function-Key Panel Editor” on page 123.
- **Disabled:** The docking of the function-key panel is not possible.
 - **As toolbar:** The function-key panel becomes a standard toolbar that you can dock on the window border.
 - **Wrap Toolbar:** If there are too many keys on the panel, they're arranged in more than one line or column.
 - **Dock as Panel:** The function-key panel becomes a toolbar, but the layout of the keys remains unchanged.

Function-key panel display

By default, the function-key panel is not displayed in the emulation window. You can select the display option from the panel configuration box (select the Show function-key panel check box).

Note:

If the width or the height of the function-key panel is greater than 1/8th of the width or the height respectively of the screen, it can't be docked.

Synchronous emulation

Choose Key Panel from the Settings menu.

Select the function-key panel which contains the function-key panel definition. Click on the Browse.... button if the function-key panel does not appear in the proposed default list and look for the function-key panel of your choice.

Note:

The function-key panel files are text files with extension .pan.

Docking

The function-key panel can be transformed into a toolbar. To do this, select the docking option you want to apply to the function-key panel: docking as a toolbar or as a function-key panel.

Show function-key panel

By default, the function-key panel is not displayed in the emulation window. You can select the display option from the panel configuration box (select check box Show panel).

Mouse configuration (asynchronous emulation)

Associating actions with one or more mouse events enhances the more traditional use of the mouse. Events are the simple click and double-click on one or two of the three mouse buttons (left button, right button and middle button on certain mice).

Escher asynchronous emulation enables a series of actions to be assigned to the two or three buttons, actions which will be executed when these buttons are simple or double-clicked.

> Configuring mouse events

Select the Display Settings... option on the Settings menu, and then click the Mouse tab.



The left window proposes the various mouse events: simple or double-click on the left button, the middle button if there is one, and the right button.

Each of these events can be associated with one or more instructions which will run as a sequence unless one of them is blocking (see list below). In this case, the following ones will not be executed. The list of instructions appears in the right window for each mouse event selected in the left window.

Note:


The actions associated with a double-click on a mouse button are executed following those associated with a single click, if there are any. This is standard Windows procedure.


Adding/deleting an event



By default, the six possible events with a three-button mouse are proposed. You can however delete an event from this list by selecting it and then clicking on the  button in the left window. To add one when the list is incomplete, click on the  button on the right window.

Adding/deleting/moving an instruction

First select the event in the left window.

To add an instruction, click on the  button in the right window. From the drop-down list which appears, select the type of instruction you want to add and if necessary fill out the right-hand edit field of the drop-down list.

To delete an instruction, click on the  in the right-hand window.

To move an instruction, use the  and  buttons to move it down or up.

Sending a string

Mouse activation according to the selected event sends a string as if it had been entered on the keyboard. Enter this string in the editing field to the right of the list of instructions.

Running a local script


If you want to write a small script to assign to the keyboard key, select the Escript option in the left drop down list:

Click the  button in the right edit field to launch the script editor.

Running a macro

Note:

This feature enables you to use the macro files created with the macro language in versions 9.0 and earlier of Tun (Esker proprietary language). The macro files are text files with extension .mac.

Mouse activation according to the selected event triggers running of a macro with format .mac. In the right-hand editing field, enter the macro file path (.mac) or select it by clicking on the  button.

Function keys

Mouse activation according to the selected event sends the value of a function key.

Select the function key of your choice from the right-hand drop-down list:

- The labels proposed are those listed in the emul.fky file, copied into the emulators installation directory.
- The values assigned to the various function keys are listed in the function keys file available for each type of emulated terminal.

Note:

The function key files are text files with extension .fun.

> For more details on the function keys, see “Association of a function key” on page 89.

Other functions

The other available functions are as follows:

- Send mouse events to host: mouse activation according to the selected event will have a meaning specific to the application itself. This action is blocking if the mouse support is activated in the application.
- Send word at cursor / Send character at cursor: the word/character on which the mouse event takes place is sent as if it were entered directly on the keyboard.
- Context menu: where applicable, the context menu concerning the screen area clicked by the mouse (this instruction is often associated with simple click on the mouse right button). This action is blocking.
- «No menu» mode menu: when the menu bar is hidden, the list of menus is displayed during the mouse event. The menu actions can then be accessed. This action is blocking.

- Normal selection: the mouse event enables a screen area to be selected while holding the button pressed down during selection (instruction often associated with a simple click on the mouse left button). This action is blocking.

Printing

Screen print

Use this option to print the contents of an emulation session. You can use Text mode (faster), or Graphic mode, which transforms colors into different shades of gray.

> **Printing a screen**

Select the Print option from the File menu, and then the print mode:

- Text mode.
- Graphic mode.
- With template in the case of 3270 or 5250 emulation.

If you print in text mode in the case of asynchronous emulation, you can choose the font used for printing. To do that, select the Use the selected font check box from the dialog box that opens and click Font... to choose the font.

In the case of graphic mode printing, you can print your emulation screens using a white background, in order to save ink. To do so, select the Force black printing on white background check box.

Configuration of printouts from the server

You can choose the printing parameters for jobs coming from the server: printer to be used, text or graphic type of printing, choice of font.

Asynchronous emulation

Choose the Print Setup option from the File menu.

Select the printer on which you want the printouts to be made. Click on the Specific printer button to configure the printer.

In certain emulations which are to include transparent printing, it can be of interest to set a wait time before closure of the print spool. This prevents the connection being closed after each data flow sent. Enter a value (in ms) for the wait time in the Print spool timeout (ms) field.

Select the Manage transparent printing check box if you want to configure transparent printing:

If necessary, select the Specify the number of lines check box and enter the number of lines you want to print on a page. Do the same with the Specify the number of columns option. This option can for example be of interest if you want to force printing over 132 columns.

You can define the font used to print in text mode. You can set the font you choose as the default print font for all printings (including that requested from the Print option of the File menu) by selecting the Use the selected font check box. Click Configure font... to specify the font to use and its style and size. Select Use the selected font size to set the font size as the default print size (if you don't select this check box, the next nearest size is used for printing).

Note:

The Script drop-down list box in the font definition dialog box lets you choose the character set encoding used by the font.

Synchronous emulation

Choose the Print option from the Settings menu.

Select the printer of your choice from the drop-down list. As in any other Windows application, the locally defined printers are available on it.

Choose Setup printer... to configure the printer for printing screens.

Select the printing mode: Print the screen in text mode (faster) or Print the screen in graphic mode.

You can define the font used to print in text mode. You can set the font you choose as the default print font for all printings (including that requested from the Print option of the File menu) by selecting the Use selected font check box. Click Setup font... to specify the font to use and its style and size. Select Use size of selected font to set the font size as the default print size (if you don't select this check box, the next nearest size is used for printing).

Note:

The Script drop-down list box in the font definition dialog box lets you choose the character set encoding used by the font.

Printing with template (3270/5250 emulations)

You will frequently want to print several or all of the pages, during a synchronous emulation session. When there are many pages, this job quickly becomes tedious and time-consuming.

To avoid this, the IBM 3270 and 5250 emulators from Esker offer printing with a template, the principle of which is to create a reusable print module.

A print template contains:

- The screen area to be printed.
- Initial and final texts markers for printing.
- Page scroll keys.

> Creating a print template

Select the Printing template option from the Settings menu.

Loading a template

Click on the Load... button to open an existing print template.

Note:

The print template files are text files with extension .ptp.


Comments

Enter an optional comment.

Print area

Enter the coordinates for the screen area to be printed in the order top left and bottom right. Click the screen at the top left-hand corner of the area you want to print and note the cursor coordinates on the right-hand side of the status bar. Click the bottom right-hand corner of the area to print and note the coordinates. Enter them in the dialog box.

Note:

The coordinates can be entered more simply by clicking the position icon  (the square formed with a dotted line) and dragging the cross-hairs to the required position on the emulation screen and releasing the mouse button. The dialog box automatically disappears while you position the cross-hair cursor. It reappears once you've chosen the position. That way, you don't have to move the dialog box to see the position you want to choose on the screen.

Also, you can use the Full Screen button to apply the coordinates of the full screen.

Previous page key / Next page key

Enter the Previous Page Key and the Next Page Key in the appropriate fields. These are system-dependent and are often shown at the bottom of the emulation screen where the commands are displayed. If they're not shown at the bottom of the emulation screen page, you should consult the documentation for the type of host you're connecting to. The codes for these keys are sent to the host to scroll from one page to another. If these aren't entered, only the current page is printed.


Note that the system might require that an alphabetic key be pressed with the Enter key. In this case, use the extra field provided for this purpose (the first field).

Return to initial page after printing

If the Return to Initial Page After Printing check box is selected, the program returns to the initial page when printing is completed.

Initial text / Final text

In the Initial Text and Final Text fields, enter the text which the emulator is to consider as marking the first and last pages to be printed. In the respective Search Area boxes, enter the coordinates of the screen zone where the emulator is to search for the initial and final

character strings. If you want to, you can use position icon .

> **Printing with a template**

Select the Print option and then With template from the File menu.

The Print with Template dialog box which opens has the look and behavior of a normal Windows print dialog box with one or two differences.

Instead of a print selection option, there is a Current Page to End option which prints from the current page to the finishing text indicated by the template. This can be the end of the file or an intermediary point.

Also, you can choose a template when you print by using the Browse... button in the Printing Template section and loading a template file (extension .ptp).

The printout consists of the emulation screen pages without any unnecessary emulation screen information from outside the print area.

Transparent printing (asynchronous emulation)

In the character flow sent to the terminal emulator, the server can insert escape sequences to inform the emulator that the following characters are to be:

- Either displayed on the screen (conventional terminal emulation).
- Or sent directly to the printer connected to the PC emulating the terminal

This functionality lets the centralized application access the printer without using a buffer or print server.

Esker supplies a series of actions used to reroute the characters received from the server either to the PC screen, or to the printer, or to both simultaneously. To work in transparent print mode, the emulator must have escape sequences associated with these actions in the escape sequences configuration file.

> See “Transparent printing” on page 157.

Cut & Paste

The Esker emulators allow the use of the clipboard for copying, cutting and pasting characters. This function can be of use both in the emulator (for example to send the copied text to the communication channel) and outside the emulator (for example to copy a part of the screen into a word processor).

> **Copying text to the clipboard**

Select the text area you require using the mouse (several lines and several columns are possible), then select the Copy option from the Edit menu or the context menu displayed using the right-hand button on the mouse.

> **Pasting the clipboard contents**

Place the cursor at the required point and select the **Paste** option from the Edit menu or the context menu displayed with the right-hand button on the mouse.

In asynchronous emulation (UNIX), to send the clipboard contents directly to the communication channel, without having to hit Enter, select the Paste and send option.

> Clear selection (IBM synchronous emulation)

Select the text area to be cleared using the mouse then select the Delete option from the Edit menu.

Copy options (asynchronous emulation)

You can set the copy options in asynchronous emulation (UNIX). To do this, select the Copy options option from the Edit menu.

- Rectangular Selection: the selected area is rectangular if this option is chosen, otherwise the lines situated between the starting and end points are complete.
- LF at end of lines/CR at end of lines: these options stipulate whether or not LF or CR characters are to be inserted at the end of the lines.
- Wait time: to prevent overloading the communication channel when cutting/pasting a large amount of text, the Wait time field enables the operation to be delayed by defining the time between copying of two blocks of 128 characters from the clipboard to the emulation window.

File transfer

File transfer in terminal emulation enables files to be exchanged between the PC and the server. This can be particularly useful, for example in order to use the content of a file in a word processor or share the files with other users.

Files can be transferred between a PC and a server:

- via FTP (standard file transfer protocol): in this case, you must have an FTP client module on the PC (this is the case with Tun).
- from an emulation session: this is proposed by the Esker asynchronous and IBM 3270 emulators.

> Configuring file transfer

Select the Configuration option on the Transfer menu.

> Sending a file

Select the Send option on the Transfer menu.

> Receiving a file

Select the Receive option on the Transfer menu.

> Canceling a transfer

Select the Cancel option on the Transfer menu.

> Transferring several files (in IBM 3270 emulation)

Select the Batch option on the Transfer menu.

Asynchronous emulation

The file transfer protocols available in asynchronous emulation are as follows:

- ASCII
- RTUNPLUS (Esker's proprietary protocol)
- X-Modem
- Y-Modem
- Z-Modem

ASCII

This is the most basic type of file transfer.

Receiving consists of capturing the characters sent across a connection in a file.

Sending consists of emptying the contents of a PC file onto the network connection, but doesn't provide any means of controlling reception. The host is responsible for capturing the data into a file.

The command most frequently used on hosts for this is:

```
stty -echo ; cat >/tmp/fichier; stty echo
```

Rtunplus

Esker's proprietary file transfer protocol, RTUNPLUS, has the advantage of being simple to use, and freely installable on any UNIX host. RTUNPLUS is provided in UNIX executable format for SCO UNIX and XENIX, and the source code (rtunplus.c) can be compiled on other hosts.

> To install rtunplus, refer to the *Getting Started* manual.

X, Y and Z-Modem

These protocols are those normally installed on most BBSs (Bulletin Board Systems). They are general-purpose file transfer protocols for use with telephone links.

The server programs for these protocols are usually not delivered as standard on most UNIX systems, but are generally available in the public domain as executables or source files for compilation.

Select the protocol you want to use from the Protocol drop-down list. Various fields will then appear.

For more information on each protocol, please refer to the online help.

Note:

The file reception box is similar to the transmission box, apart from a few differences concerning the transfer direction: for example, the Local source file field becomes Host source file

Source File

Enter the source file name in the Name field.

Target file

Enter the target file name.

Text file: convert...

This option converts the contents of the file from the local character set to the character set of the remote machine. The Conversion button displays a dialog box that lets you select the two character sets.

CR/LF -> LF

This option applies or cancels CR/LF to LF (or LF to CR/LF) conversions. It's useful when transferring text files from a PC to a UNIX server or vice-versa.

Use Protocol

Specify one of the available transfer protocols.

Options

Use the Options button to select a protocol and set related parameters.

If ... File Exists

This option determines what action to take if you transfer a file that already exists on the destination machine.

Send

Click this button to begin the transfer procedure. If you're using RTUNPLUS, then the UNIX host must be in the shell (with the #, %, or \$ prompt on the command line) to run the server portion of the protocol. To use X, Y and Z-modem, you may need to manually start the server on the UNIX host (that is, with a command such as `xmodem -r /tmp/tmp.file`).

3270 synchronous emulation

File transfer in 3270 synchronous emulation uses the IBM IND\$FILE protocol.

Host System

Defines the type of host operating system you're connecting to.

Timeout

Indicates the length of time the program must wait for a host response to the file transfer request. The default is 30 seconds.

Packet Size

The default packet size is 2040 bytes. The speed of the file transfer is directly proportional to the size of the packet.

Host Command

Indicate the command that is to handle the transfer. The default is IND\$FILE.

Initial Action

Defines the initial action to be performed before the transfer starts.

ASCII/EBCDIC Conversion

Choose the appropriate option for the file transfer you're preparing.

- Host Performed Conversion means characters are converted to the host code page by the host.
- Use Current Character Set converts received characters to the character set loaded: This is performed by the PC.
- You can select a different character set by browsing the .set files, thus making allowance for the differences in national character sets.

Note:

The following fields are those accessed via the Browsing tab.

Host command

Enter the command sent to the server to retrieve the file list when performing a transfer. By default, the FILELIST command is used for a CMS system and LISTCAT for a TSO system.

Initial action

If necessary, select the initial actions that precede the command.

Note:

The file reception box is similar to the transmission box, apart from the configuration of the save format. You can switch from one to the other by selecting the Send or Receive options in these boxes. This is particularly useful when transferring a number of files.

Remote file

Enter the name of the file on the server.

Local file

Enter the name of the local file (absolute path).

You can use the Browse button to search for the files.

System

Select the server system (CMS, TSO, CICS).

Conversion

Set the conversion type: Select the ASCII/EBCDIC check box to perform the conversion to EBCDIC, the CR/LF check box to replace CR/LF codes with end of record codes, and the JISCI check box to perform the conversion from Shift JIS to EBCDIC (the NOSO (SO/SI code) check box should then be selected if the file only contains DBCS characters).

Mode

Select the copy mode (replace existing file or add to this file).

Record format

For a file transfer to a TSO or CMS server, select the unshaded options if necessary.

The default record format means that the record length complies with the server's default values. If you don't want this to be the case, select the Variable option or the Fixed option for CMS and TSO environments. You can also use the option Undefined for the TSO environment.

For TSO environments, select the unit: Default to use the server's default unit, otherwise Cylinders, Tracks and Blocks. Enter the block size value, and the track and cylinder values if you select these units as space media.

In any case, consult the documentation concerning your particular server environment to find out about the options applicable to your system.

Additional options

This field lets you add parameters for the IND\$FILE program that haven't been entered elsewhere in the dialog box.

Multiple file transfer (synchronous emulation)

Multiple file transfer enables you to specify the list of files to be exchanged between the PC and the server once you start the transfer operation.


When transferring multiple files, a dialog box keeps you informed on the status of each transfer. You can save the contents of this dialog box in a log file, and then view it later on.

> Transferring several files


Select the Batch option on the Transfer menu.



List of transfers

Each file on this list corresponds to a file to be sent or received, with its transfer parameters.

To add a file to the list for transfer, click on the  button. Then fill out the file transfer parameters in the same way as for a simple transfer (see the description of transmission and reception boxes in the File transfer/Synchronous emulation section).

To change the transfer parameters of a file, double-click on this file (see description of transmission and reception boxes in the File transfer/Synchronous emulation section).

To delete a file from the list, select the file and click on the  button.

To define the order of transfer for a file on the list, select the file and click on the  and  buttons.

Wait time

The Time to wait field enables you to define a waiting period between transfer of two files on the list.

Start transfer

Click on the Transfer button to start the multiple transfer operation.

Save and load

You can save your list of files to be transferred by clicking on the Save button. Then in the save box which appears, give the file name and path (extension .xfr), and click on OK.

You can then reload this list of files to be transferred by clicking on the Load button and selecting your transfer file (extension .xfr).

> Saving in a log file

Select the Log option in the Settings menu.

Select the Use a log file option to save the contents of this dialog box in a log file (this option is not selected by default).

Enter both path and name of the file to be used (file extension .log) in the Log file field, or click Browse... to select a file.

Display of controls after recognition of a string

A traditional application can be enhanced with controls (buttons, menus, customized texts, combined lists, scrollbars) automatically positioned on the screen when a particular character string appears on it. Identification of the string and positioning of the controls are two consecutive operations managed by the active zone (or hotspots) function. .

> Configuring the hotspots

To access configuration of all the hotspots, select the Hotspots... option on the Settings menu.

Or use the mouse to select the character string to be recognized on the screen, click on the mouse right button to call up the context menu and select Create a hotspot. The control created and the hotspot selected are then added to the list of hotspots accessible from the Hotspots... option on the Session menu.

> See “Hotspots ” on page 112.

Terminal customization

A terminal has its own configuration parameters (for example, cursor size, special character sets, etc.). You can change these parameters if you so wish.

> **Configuring the terminal**

Select the Terminal option on the Settings menu.

Asynchronous emulation

When you choose a type of terminal at connection, the existing terminals have a corresponding configuration file associated with the session. This file contains the terminal configuration, the main functions of which you can then modify.

Note:

The terminal configuration files have extension .ses. The parameters that can be modified here are those declared in the [Intro] section of these files. The content of the above dialog box therefore depends on the content of the file.

> See “Content of a terminal configuration file” on page 146.

Synchronous Emulation

Print Key

In 3270 emulation, the Print key option can be ignored (first option) or used to print according to the local print mode (Print option of the Session menu).

In 5250 emulation, the Print key option can be used as it normally is on a 5250 terminal, or activated according to the local copy option (Print option of the Session menu).

Ctrl key

Select either or both of these options to validate the redefinition of this key. You redefine the Ctrl key by choosing Keyboard in the Session menu.

Scope of the Caps Lock key

Caps Lock Definition determines whether the Caps Lock key is allowed to effect the whole keyboard or simply the main alphanumeric keyboard, excluding the numerical keypad. In this way, the symbolic keys on the keypad, Up, Down, Home, etc., are freed for other purposes.

Other options

You can also choose to use the keyboard in advance, by defining the size of the buffer used for the keyboard.

You might find it useful to select the Beep for Operator Error check box. For example, if you press an alphanumeric key when the cursor isn't in the right place, you're alerted by a beep.

Note:

The fields described below are those of the Session tab.

Cursor style

Define Cursor style: Underscore, Block or Vertical Bar, Solid or Blink.

Show Rule Cursor

Note that if the Show Rule Cursor check box is selected, the line at which the cursor is positioned is underlined to facilitate, for example, the viewing of a dense text file.

Numeric Fields Support (3270 emulation)

In 3270 emulation, select the Numeric Fields Support check box if you want to restrict user input to numeric characters. If the check box is left unselected, any alphabetical character can be entered in numeric fields.

Moving the cursor at first mouse click

When the Esker Viewer window is inactive because of another application activity, right-clicking the Esker Viewer window can generate two different situations:

- The Esker Viewer window turns active again and the cursor moves under the mouse pointer (where the click occurred). For this, select the Move the cursor at first mouse click check box.
- The Esker Viewer window turns active again, but the cursor stays at its previous position. You will need to click again on the emulation window to move the cursor under the mouse pointer.

Note:

The fields described below belong to the HLLAPI tab.

Short name

Select a letter to associate a short name to the session.

Assigns automatically

Select the Assigns automatically check box to let the emulator self-assign a HLLAPI name to the session.

Long name

Enter a comment in this field by way of a long name for the session.

Modifying the character table (IBM synchronous emulations)

Synchronous emulation uses the EBCDIC extended character set which can vary from country to country. If the current table doesn't meet your needs, you may change it. .

> Modifying the character table

Select the Character set option from the Settings menu.

The current EBCDIC table is shown on the right. It presents the characters displayed on the screen. The character set available on the PC is on the left. Choose between the ANSI and OEM character sets. You can only modify the EBCDIC table.

You can:

- Load the table corresponding to your language by clicking on the Load button and selecting the .set file of your choice.
- If necessary, select from the left-hand table (PC characters) the special character you want to display in place of another. Then drag it into the right-hand table over the character it is to replace.

Click **Save** to save the modified character table.

Customizing the keyboard

Esker supplies its emulators with a default keyboard configuration. You can however change this configuration by customizing each of the keys of your PC so that a different result from that proposed by default is obtained when it is struck. Depending on the type of emulation, specified below, you can associate the following with a PC keyboard key:

- The value of a terminal keyboard key.
- A character or character string.
- A function key.
- A function related to the type of terminal.
- A macro or a script.

> Customizing the keyboard

Select the Keyboard... option on the Settings menu.

The configuration box shows two keyboards:

- The PC keyboard at the top.
- The emulated terminal keyboard at the bottom.

Asynchronous emulation

Keyboard selection

Each type of emulated terminal has a corresponding default keyboard configuration. This default configuration is in fact an association between a PC keyboard and a terminal keyboard.

Example:

If you are in a VT100 emulation session, the keyboard of a classic VT100 terminal will be displayed in the lower part of the keyboard configuration box.

However, you can select another type of keyboard for the PC and for the emulated terminal: select the type of keyboard you want from either of the drop-down lists proposed in the configuration box (PC Keyboard and Terminal Keyboard).

Notes:

The PC keyboard configuration files take extension .kbn. The terminal keyboard configuration files take extension .kbt.

Only the files with extension .kbn and .kbt present in the emulator's installation directory will be available in the drop-down lists.

PC keyboard

Each of the keys on the PC's keyboard has an associated scan code. When you press a key you send the code over the communication channel. For example, when you press the key e, you send the code 18.

The code is interpreted by the keys configuration file, associated with the type of terminal emulated. For a given scan code, eight values are possible: base key, key combined with the Shift, Alt, Ctrl, Ctrl Shift, Alt Shift, Alt Ctrl, Ctrl Alt Shift keys..

Note:

The key configuration files are text files with extension .key. They contain one section per keyboard key, with each section comprising the values for the various possible combinations of the key.

The Lock state drop-down list specifies the key behavior in relation to the Caps lock and Num lock keys.

If Caps lock mode is enabled, the key pressed in Caps lock mode will be considered as shifted. This is often the case with characters. If Num lock mode is enabled, the key pressed in Num lock mode will be considered as shifted. It is often the case with Numeric Key Pad keys. If No is selected, both Caps lock and Num lock won't have any effect.

Sometimes the keyboard characters are language-specific. The scan code mnemonic is then nat, which means that the code sent must be interpreted by a national file.

Note:

The national files are text files with extension .nat. They can replace the keyboard configuration files when a national keyboard is specifically loaded into the emulation.

Terminal keyboard

The dialog box displays the values and actions generated for the different states of the keys on the selected terminal keyboard: base, shift, ctrl, ctrl shift, alt, alt shift, alt ctrl, alt ctrl shift. You can also see the values returned by the keys of the current terminal keyboard and use these values for your PC's keyboard. These values cannot be modified.

Methods for assigning a value to a PC keyboard key

Very simply, to assign the value of an emulated terminal keyboard key directly, you should use the drag-and-drop method:

Display the emulated terminal keyboard if not already done, by selecting the Terminal Keyboard option.

Select the key to be assigned on the terminal keyboard.

With the mouse, drag the terminal keyboard key to that on the PC keyboard.

Otherwise, to assign a particular value to a PC keyboard key which has been pre-selected, display the actions editor by selecting the Actions list option.

The left-hand drop-down list can be used to select the type of value assigned to the key: character string, scripts, .mac type macros, function keys, mnemonics. On the right, you can type in or select from a list the value assigned to the key.

The following paragraphs concern the use of the actions editor which you should call up first as explained above.

Sending a character string

From the actions editor, select the String option from the left-hand drop-down list.

For a simple character, in the right-hand editing field, type in the character as you want it to be sent when the key is struck:

- For a printable character: type in the character directly with or without quotation marks (example : a or “a” or 'a' will send letter a).
- For a decimal character: type in the decimal value of the character, between 0 and 255 (example: 97 for a, 63 for ?,...).
- For an octal character: type in the octal value of the character, between 0 and 0377, preceded by a 0 (example: 0141 for a, 077 for ?,...).
- For a hexadecimal character: type in the hexadecimal value of the character, between 0 and FF preceded by 0x (example: 0x61 for a, 0x3f for ?,...).

Note:

The ' character is coded by '\ ' and the \ character by '\\'

> You can also enter a mnemonic symbolizing a simple character. See “Association of a mnemonic” on page 90..

If you want a key to emit a character string, enter the characters (between quotes and in the correct order) on the desired key in the right-hand editing field. If a character is not print-

able, use decimal, hexadecimal, or octal notation, preceded by the character \. Mnemonics aren't recognized in character strings.

Examples of character strings:

| Value | Result |
|---------|---|
| aef | string aef displayed on the screen |
| a\033be | character a, escape and character b not displayed, then character e (final result: string ae displayed on the screen) |
| a\0x08b | character a, backspace (bs) on the a, then character b (final result: character b displayed on the screen) |
| \32i\10 | space (sp), character i and linefeed (lf) |

Running a script

If you want to write a small script to assign to the keyboard key, select in the actions editor the Escript action in the left drop down list:


Click the  button in the right edit field to launch the script editor.

Running a macro

Note:

This feature enables you to use the macro files created with the macro language of the previous Tun versions (Esfer proprietary language). The macro files are text files with extension .mac.

From the actions editor, select the Macro option from the left-hand drop-down list.

In the right-hand editing field, enter the macro file path (.mac) or select it by clicking on the  button.

Association of a function key

From the actions editor, select the Function option type from the left-hand drop-down list.

Select the function key of your choice from the right-hand drop-down list:

- The labels proposed are those listed in file emul.fky, copied into the emulators installation directory.
- The values assigned to the various function keys are listed in the function keys file in each type of emulated terminal.

Note:

The function key files are text files with extension .fun.

> See “Advanced Use of Asynchronous Emulator” on page 137 concerning the function key files.

The emul.fky file comprises one section per type of terminal, containing the labels of from one to one hundred function keys.

```
[ansi]
fKey1=F1
fKey2=F2
fKey3=F3
fKey4=F4
...
fKey57=End
fKey58=Down arrow
fKey59=Page down
fKey60=Inser

[...]

[vt100]
fKey1=F1 PF1
...
fKey5=F5 Break
fKey6=F6 F6
...
fKey57=End PGUP
fKey58=Down arrow
fKey59=Page down PGDN
fKey60=Insert FIND

[...]
```

You can modify this file in order to call up the labels of your choice, symbolizing the function keys of the emulated terminal.

Association of a mnemonic

Mnemonics are short words interpreted by the emulator to perform a particular action.

In the actions editor, choose the Mnemonics option from the dropdown list on the left.

The mnemonics of the Esker asynchronous emulator can symbolize:

- A simple character (example: nul to send 0).
- An action (example: nlock to activate the NumLock key to lock the numerical keypad).

- A dead key, that is a key which only has an effect after another key is pressed (for example: tilde to add a ~ to a letter).

Here are three lists of mnemonics:

Simple characters

| Mnemonics | decimal | octal | hexadecimal |
|------------------|----------------|--------------|--------------------|
| nul | 0 | 00 | 0x00 |
| soh | 1 | 01 | 0x01 |
| stx | 2 | 02 | 0x02 |
| etx | 3 | 03 | 0x03 |
| eot | 4 | 04 | 0x04 |
| enq | 5 | 05 | 0x05 |
| ack | 6 | 06 | 0x06 |
| bel | 7 | 07 | 0x07 |
| bs | 8 | 010 | 0x08 |
| ht | 9 | 011 | 0x09 |
| lf | 10 | 012 | 0x0a |
| nl | 10 | 012 | 0x0a |
| vt | 11 | 013 | 0x0b |
| ff | 12 | 014 | 0x0c |
| np | 12 | 014 | 0x0c |
| cr | 13 | 015 | 0x0d |
| so | 14 | 016 | 0x0e |
| si | 15 | 017 | 0x0f |
| dle | 16 | 020 | 0x10 |
| dc1 | 17 | 021 | 0x11 |
| dc2 | 18 | 022 | 0x12 |
| dc3 | 19 | 023 | 0x13 |
| dc4 | 20 | 024 | 0x14 |
| nak | 21 | 025 | 0x15 |
| syn | 22 | 026 | 0x16 |
| etb | 23 | 027 | 0x17 |
| can | 24 | 030 | 0x18 |
| em | 25 | 031 | 0x19 |

| Mnemonics | decimal | octal | hexadecimal |
|------------------|----------------|--------------|--------------------|
| sub | 26 | 032 | 0x1a |
| esc | 27 | 033 | 0x1b |
| fs | 28 | 034 | 0x1c |
| gs | 29 | 035 | 0x1d |
| rs | 30 | 036 | 0x1e |
| us | 31 | 037 | 0x1f |
| sp | 32 | 040 | 0x20 |
| del | 127 | 0177 | 0x7f |
| ind | 132 | 0204 | 0x84 |
| nel | 133 | 0205 | 0x85 |
| ssa | 134 | 0206 | 0x86 |
| esa | 135 | 0207 | 0x87 |
| hts | 136 | 0210 | 0x88 |
| htj | 137 | 0211 | 0x89 |
| vts | 138 | 0212 | 0x8a |
| pld | 139 | 0213 | 0x8b |
| plu | 140 | 0214 | 0x8c |
| ri | 141 | 0215 | 0x8d |
| ss2 | 142 | 0216 | 0x8e |
| ss3 | 143 | 0217 | 0x8f |
| pu1 | 145 | 0221 | 0x91 |
| pu2 | 146 | 0222 | 0x92 |
| sts | 147 | 0223 | 0x93 |
| cch | 148 | 0224 | 0x94 |
| mw | 149 | 0225 | 0x95 |
| spa | 150 | 0226 | 0x96 |
| epa | 151 | 0227 | 0x97 |
| csi | 155 | 0233 | 0x9b |
| st | 156 | 0234 | 0x9c |
| osc | 157 | 0235 | 0x9d |
| pm | 158 | 0236 | 0x9e |
| apc | 159 | 0237 | 0x9f |

Actions

| Mnemonic | Action |
|----------------------|--|
| nop | No action |
| lshift | Activates left shift key |
| rshift | Activates right shift key |
| ctrl | Activates control (Ctrl) key |
| alt | Activates Alternative (Alt) key |
| clock | Activates Caps Lock key |
| nlock | Activates Num Lock key |
| slock | Activates Scroll Lock key |
| cal0.....cal9 | Successive pressing of numerical keys to obtain the corresponding decimal code (such as <Alt> 1-2-3 in MS-DOS) |
| hdcopy | Hard Copy of the screen |
| scr1 | Switches to session 1 |
| scr2 | Switches to session 2 |
| scr3 | Switches to session 3 |
| scr4 | Switches to session 4 |
| altpg1...8 | Switches to the specified page (on multi-page terminals) |
| nscr | Switches to the next session |
| send | Sends Windows file (Alt-F7) |
| receive | Begins reception of a Windows file (Alt-F8) |
| freceive | Ends reception of a Windows file (Alt-F9) |
| brk | Sends a break signal to the host |
| femul | End of emulation (Alt-F10) |
| win | Returns keyboard control to Windows |

Dead Keys

| Mnemonic | Example |
|----------|---------|
| acute | é |
| grave | è |
| cflex | â |
| tilde | ñ |
| trema | ï |
| ring | Å |
| cedil | Ç |
| bar | ç |

Lock state

The Lock State check box indicates the key's reaction to <Caps Lock> and <Num Lock>.

Click the check box to switch from one lock state to another.

If the check box has the label Caps lock, the key can be shifted when Caps lock is on. Typically, this applies to characters. If the check box has the label Num lock, the key can be shifted when Num lock is on. Typically, this applies to the keys on the numerical keypad. If the check box has the label No, Caps lock and Num lock have no effect.

Canceling changes

To cancel the changes you make to a scan code and apply the previous value, click the Scan code section Cancel button. The cancellation only applies to the current changes: If you click another key without canceling, the changes are accepted.

To cancel changes made to the keyboard since the last save, click the Cancel button on the right of the keyboard.

Saving the modified keyboard

Click Save or Save As to save the modified keyboard to file.

Note:

The keyboard configuration files in asynchronous emulation are text files with extension .key.

Synchronous emulation

Click on a PC keyboard key to identify the value sent when this key is struck, while reading the content of the unshaded edit field.

You can also find out the value of a key combined with the Shift, Ctrl and Alt keys: activate the base key and the combination key, then read the content of the unshaded edit field.

Note:

It is only possible to combine with the Ctrl key on the left or right of the PC keyboard, if the Left

Ctrl key and/or Right Ctrl key check boxes have been selected in the Keyboard tab of the Options dialog box.

To assign a new value to a key (or combination of keys) on the keyboard, you can:

- Attribute the value of a key from the real terminal keyboard to the key.
- Associate one or more actions to this key (character string, IBM function, macro, script).

A key whose scan code has changed appears in dark gray on the PC keyboard

Note:

To reassign its default value to a key, click the key with the right mouse button to display the context menu. Select the combination you want and click Default. In the same way, to delete the key value, choose Clear from the context menu.


Assigning a terminal keyboard's key value

To assign a real terminal keyboard's key value to the PC keyboard key:

1. Select the key you want to change on the PC's keyboard as well as the combination key, if any.
2. Select the key on the terminal's keyboard you want to assign as well as the combination key, if any.
3. Drag and drop the terminal keyboard key to the PC key position.


Assigning actions


To assign one or more actions to a key (function, character string, or macro), first double-click the key and select Shift, Alt or Ctrl to define the key combination if necessary. You can also double-click in the edit field, which appears unshaded.

To add an action, click the  button. Select the action you want to assign to the key from the list box: Function, String, Macro, local script in one of the three available languages (VBScript, JScript, EScript), script functions library.

Function: Select the function you want from the list box.




String: Enter the character string in the field on the right.

Macro: Enter the file path for the macro in the field on the right or select a macro file using the browse button .

Local script: Select the EScript option and click the  button in the right edit field to launch the script editor and write your script.

Note:

The order in which you enter the actions is the order in which they're executed when the key or combination of keys is pressed.

To delete an action from the list, select it and click the button . To move an action up or down in the list, select it and click the button  to move it up the list, or  to move it down.

Save / Load keyboard

Click Save... to save the modified keyboard. Click Load... to load an existing keyboard.

Note:

The keyboard configuration files in synchronous emulation are text files with extension .kbd.


Macro automation

You can execute, record or edit a macro from the Esker emulators. Macros created this way can be included in a function-key panel, or associated to a connection startup/end, a keyboard key, a hotspot...


Recording macros

You can create simple macros by choosing Macro > Record Macro or clicking the toolbar button .

Enter the name of the macro you want to create in the dialog box that opens. Click Record to start recording the macro. Each action you perform is recorded in the macro.

To stop recording the macro, choose Macro > Stop from the main menu or click the toolbar button  again.

Characters entered from the keyboard are translated into a Send instruction.

To generate a Receive instruction (wait for a character string), select one or more lines with the mouse. Then choose Macro > Wait for selection or click the button  in the toolbar. Each line is then associated with a Receive instruction and an accompanying IfError ERROR instruction which generates an error if the string isn't received. The default wait time is 10 (the instruction is Receive 10 string).

A pause (over 4 seconds) generates a Sleep time instruction in the macro, where time is the duration in seconds of keyboard inactivity.

Password encryption in macros

The Send and SendAndReceive commands let you send character strings to the server. If you're sending a password, it's important to be able to make it invisible in the .mac macro file.

There are two ways to encrypt a character string:

- Use an external authentication library (passlib.dll). To do that, you must have a copy of the library on your machine: You can then select the Use External Authentication Library check box from the Settings > Macros... option.

- Use the encryption algorithms supplied with the Tun range. In this case, clear the Use External Authentication Library check box from the Settings > Macros... option.

If you want to encrypt the strings sent to the server, you can choose Macro > Encrypt Transmissions.


For example, if you want to create a login macro with an encrypted password, do the following:

1. Open an emulation session and start recording a macro (Macro > Record).
2. Deactivate encryption by clearing the Macro > Encrypt Transmissions check box.
3. Press Enter to obtain the login prompt.
4. Select login on the emulation screen, then choose Macro > Wait for Selection.
5. Enter the user name and press Enter.
6. Select password on the emulation screen, then choose Macro > Wait for Selection.
7. Choose Macro > Encrypt Transmissions to activate encryption.
8. Enter the password and press Enter.
9. Clear the Macro > Encrypt Transmissions option to deactivate encryption.
10. Tell the macro to recognize the shell prompt by selecting \$ and then choosing the Macro > Wait for Selection option.
11. Stop the recording.

If you want to copy the encrypted code for a particular character string into the .mac file, choose Macro > Encrypt String....

Enter the character string you want to encrypt. Click Encrypt. Select the encrypted string, copy it to the Clipboard (Ctrl-C), and click Close. Copy the string from the Clipboard to your .mac file (Ctrl-V) and precede it with the character @.

Editing macros


You can edit macros to check or change the contents, or set parameters to be entered from the keyboard (for example, a password). To do that, choose Macro > Edit... or click the toolbar button  when you've finished recording the macro. Select the macro you want to edit from the dialog box and click Open.



Click NIS to access a macro file registered with the NIS server.

For more details on macros, see the chapter “Emulator Automation” on page 133.

Running a macro

To execute a macro, choose Execute from the Macro menu or click the  button in the toolbar. Select the macro you want to run and click Open.



Click NIS to access a macro file registered with the NIS server.

For more details on macros, see the chapter “Emulator Automation” on page 133.


Association of a startup or end macro

You can associate a macro to the beginning or to the end of the emulation session: this allows, for instance, to automate the connection to the host (startup macro), or to properly quit an application when closing the session (end macro).

> See “Emulator Automation” on page 133 for more information on macros and scripts.

> Associating a macro with the emulation session

Select the Macro option in the Settings menu.

Select a macro (file extension .mac) containing a unique macro written in Esker's proprietary language (compatible with previous versions of Tun). Enter the macro file name in the right field. You can use the  button to search for the file in your file system.

Add any other parameter required for the execution in the Parameters field, each occurrence separated by spaces. If a parameter contains a space, you must specify this parameter between double quotes (“”).

Emulation customizer

You can adapt the asynchronous or synchronous emulators to users' needs by selecting the menus and options you want to appear in the main menu. To do that, you must define a profile which lets you generate a customized version of the application's user interface and associate it with a particular configuration.

Defining a profile

Choose Setting > Emulation Customizer

The dialog box shows a list of all the application's menus, submenus and menu options:

The menus and submenus are represented by the icon .

The menu options are represented by the icon .

Select the menus, submenus and menu options you want see displayed in the main menu:

To select all the submenus and/or options of a particular menu, click the menu's (or submenu's) check box to activate it.

To unselect all the submenus and/or options of a particular menu, clear the menu's (or submenu's) check box.

To select or clear particular menu options, click the corresponding check boxes to enable or disable them.

Notes:


A menu's (or submenu's) check box is enabled if one of the menu items (a submenu or an option) is selected. The check box is grayed if any of the options are cleared (that is, not selected).

A menu's (or submenu's) check box is deactivated if no menu items (submenus or options) are selected.


Click Apply menu customization to current configuration to apply the changes immediately to the application's menus, without exiting the customized emulation generator.

Saving a customization profile

You can save the profiles you create in new executable files and associate these executable files with the emulator's configuration files. To do that:

Complete the field Executable to create, specifying the path and name of the executable (.exe file) that will be linked to the customized menu. Click the Browse button  beside the field to select an executable file.

If you want to associate a particular archive configuration file with this executable, select the Include configuration check box. Complete the corresponding field, specifying the file path and name of the configuration file you want to use. This file can be the emulator's default configuration file (.cfz format, 3270.ini or 5250.ini, or .cfs format) . Click the

Browse button  beside the field to select a configuration file. If no configuration file is associated with the executable, it uses the emulator's default configuration file.

Click Generate... to create the new executable. To start the new program, click the corresponding .exe file.

If you want to replace the current version of the emulator, first check that any changes to the menu profile and configuration parameters will be saved on exit. Then quit the program and confirm changes if asked.

Note:

You must be careful about deleting the option Settings > Emulation Customizer from the menu bar. If you delete this option from a program, you can no longer change the menu system.

If you deleted this option from a customized program and you then want to change the program's menu system, do the following:

1. Redefine a profile from the emulation program supplied by Esker.
2. Save it with the name of your customized program.

If you deleted this option from the actual emulator supplied by Esker (emul32.exe, 3270_32.exe,

5250_32.exe) and you want to return to the original configuration (with a full menu bar), do as follows:

--In asynchronous emulation, delete the ProfileId and ProfileNId values created in the following key:
HKEY_CURRENT_USER/Software/Esker/Tun/8.00/TunEmul/Options

--In synchronous emulation, open the configuration file used by the executable (3270.ini or 5250.ini in the emulator's installation directory, or the configuration file .cfs associated with the executable). Then delete the [Profile] section (created when you customized the menus) from the file and save it.

Firewall

Connections to computers outside the local network can be put through a proxy firewall.

To set the firewall parameters (IP address, port number, etc.), choose Option > Firewall from the main menu.

To configure the gateway, select the Use a Proxy server (SOCKS protocol) check box. Enter the name or IP address of the server (only enter a name if your system uses a DNS). You can also choose the server from the drop-down list box: The list contains the servers registered in the host table (hosttab) and on the NIS server (NIS resources are shown in yellow).

Enter the port number for the SOCKS protocol (typically, this is 1080, the default value).

If you don't want to use the gateway for local network communications, select the Do not use for local addresses check box.

You can apply the firewall settings to all the Tun applications on your machine by selecting the Use these settings for all Tun applications check box. To apply the general settings to all the Tun applications (after using special settings in Tun EMUL, for example), click Restore general settings.

IBM Printers Emulation

The IBM synchronous emulation enables the printing of an IBM terminal data to a printer connected to a Windows based PC.

With the help of the IBM 3270 or 5250 terminal emulator, you only need to connect to the terminal via an emulation session on your PC. Data coming from the terminal are then displayed on the emulation screen, and can be printed using the emulator's print command.

> This type of screen printing from Esker's synchronous emulators is described in the **Using emulators/Printing** section.

Or, you can also print data from the terminal, emulating an IBM printer on your PC. Instead of being displayed on an emulation screen of the PC, the data will be printed directly on a printer connected to the PC. You will then be able to print data from your IBM applications to any printer accessible from your PC, just like with a real IBM printer.

Esker provides the IBM 3287 and 3812 printers emulation on your PC:

- The 3287 emulation to print data flows from IBM MainFrame servers.
- The 3812 emulation to print data flows from IBM AS/400 systems.

Preliminary notion: Logical Unit (LU)

The IBM SNA architecture is based on a connection oriented, hierarchical and centralized model. In this model, a terminal or a printer is designated by a LU (Logical Unit).

There are different types of LU, among which:

- LU1 and LU3, which designate printers;
- LU2, which designate 3270 terminals.

In this chapter, we will closely focus on LUs used for printing, i.e. type 1 and 3 LUs:

- LU3 is currently the most used LU type for IBM printing. It supports DSC mode data flows.
- LU1, less frequently used, supports the SCS mode, which allows to send page layout commands such as tabs, margins, page breaks, bold, italic, etc from the server.

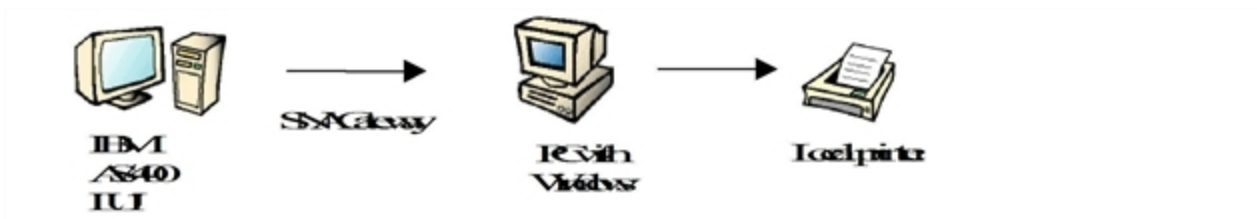
3287 Printer Emulation



The 3287 printer emulation is carried out via the opening on a MainFrame server of an emulation session identified by a LU name that corresponds to a type 1 or type 3 LU on the server.

The 3287 emulation can use the following types of connection: TN3270E connection, connection through an SNA Server or Netware for SAA gateway.

3812 Printer Emulation



The 3812 printer emulation is carried out via the opening on an AS/400 system of an emulation session identified by a LU name that corresponds to a type 1 LU.

The 3812 emulation can use the following types of connection: TN5250E connection, connection through an SNA Server or Netware for SAA gateway.

Using IBM printers emulation

The 3287 and 3812 printers emulations provided by Esker can be used in two different ways:

- From the 3270 or 5250 terminal emulation application, to associate a printing session to an open terminal session and to allow the user to print data from the terminal interactively. This type of use corresponds to punctual printing jobs from applications on the terminal to which the user is connected via an emulation session.
- From the 3287 or 3812 printer emulation application, to allow data printing from various sessions configured and centralized on the PC, which acts as a print server. This type of use corresponds to a steady and automatic printing of IBM systems data.

From the 3270 or 5250 terminal emulation application

To proceed with a punctual printing from a terminal emulation session associated to a printer emulation session, run the 3270 or 5250 emulator and connect to your terminal.

> See the **Synchronous emulation** chapter for more details on this type of connection

Select Start the associated printer in the Associated printer menu to startup the 3287 or 3812 print server: This operation allows the 3287 or 3812 application to establish a printer emulation connection according to the already configured 3270 or 5250 terminal emulation session settings.

When you start the print server, it appears as an icon in the task bar. Click on this icon to open the print server window.

From the printer emulation session you just defined, you must configure the 3287 or 3812 printing options used by your server to send data to be printed, then configure the printer connected to your PC to print data from the server.

You can then launch your print jobs from the 3270 or 5250 terminal emulation session. You can check their proper operation from the printer emulation session window.

Configuring the 3287/3812 printing options

Select Terminal in the Settings menu from the 3287 or 3812 print server.

Configuring the PC printing

From the 3287 or 3812 print server, select Print in the Settings menu.

From the 3287 or 3812 printer emulation application itself

To allow data printing from your IBM system without connecting to it from a terminal emulation session, you must start the 3287 or 3812 print server and configure the various printer emulation sessions.

Configure the 3287 or 3812 printers emulation sessions that will provide the connection to the servers, from which you will launch print jobs later on.

For each session, configure the printing options used by the server to send data to be printed, and the printer used by the PC to print data coming from the server.

Print jobs coming from various servers can then be sent to the printers connected to the PC. You can check their proper operation either from the 3287 or 3812 print server window, or by consulting the log file.

Starting the print server

Click the 3287 printer or 3812 printer icon from Access to applications in the Esker Tun program group to start the 3287 or 3812 program.

Configuring an emulation session

Select New in the File menu and specify the session connection settings.

Configuring the 3287/3812 printing options

Select Terminal in the Settings menu.

Configuring the PC printing

Select Print in the Settings menu.

Connection in IBM Printers Emulation

IBM 3287 printing emulation connection

Esker's IBM 3287 emulator enables you to make the following types of connection:

- Direct connection to a MainFrame server or via a SNA-TCP/IP UNIX gateway using the TN3270E protocol.
- Connection to a MainFrame server via a Microsoft SNA Server or Novell Netware for SAA gateway.

IBM 3812 printing emulation connection

Esker's IBM 3812 emulator enables you to make the following types of connection:

- Direct connection to a AS/400 or via a SNA-TCP/IP UNIX gateway using the TN5250E protocol.
- Connection to a AS/400 via a Microsoft SNA Server or Novell Netware for SAA gateway.

Customizing a printing session

This box contains two tabs:

- The Session tab enables you to specify the type and the parameters of the connection to be made.
- The State tab gives the state of the connection (the state is Unconnected at the moment of connection).

Options common to the Session tab

Selecting the Close Session on Confirmation check box displays a confirmation dialog box on exiting the program.

If the Startup Connection check box is selected, the program will automatically reconnect at the host.

If the Reconnection after a shutdown check box is selected, the program will automatically re-establish the connection if it's interrupted by the host.

Session tab in TN3270E connection (3287 emulation)

This section describes the Session tab as it appears in 3287 emulation when the connection type is TN3270E.

Terminal Type

This field sets the type of 3287 terminal display you want to use. Here, the only available template is Template 1, which corresponds to a printer.

Host name

Enter the name of the server or its IP address. You can also select a server from the drop-down list (only enter a name if you have a names server).

Port

The default port number is 23. You can change this number, if necessary.

Associated printer

When you define a 3287 emulation session, you must state the printer logical unit (LU) that you want to use on the host machine. Some terminal LUs are associated with printer LUs (LU1, LU3) on the host machine. Select the Associated printer check box if you want to associate a terminal LU (LU2) with a printer LU rather than simply enter a printer LU.

Printer logical unit

This field only appears if the Associated printer check box isn't selected. You can enter the name of the printer LU (LU1, LU3) that'll be used by the host machine for 3287 print emulation. The name of the printer LU must correspond to the type of terminal defined on the host machine, namely Model 1.

Terminal logical unit

This field only appears if the Associated printer check box is selected. It lets you enter an LU terminal name (LU2) that's associated with the name of the printer LU on the host machine that will be used to emulate 3287 printing. The name of the terminal LU associated with a printer LU must correspond to the type of terminal defined on the host machine, namely a Model 1 printer.

Session tab in SNA Server or Netware for SAA connection (3287 emulation))

This section describes the Session tab as it appears in 3287 emulation when the connection type is SNA Server LUA, SNA Server FMI or Netware for SAA.

Terminal Type

This field sets the type of 3270 terminal display you want to use. Here, the only available template is Template 1, which corresponds to a printer.

Printer logical unit

Enter the name of the logical unit (LU) you want to work on.

Session tab in TN5250E connection (3812 emulation)

This section describes the Session tab as it appears in 3812 emulation when the connection type is TN5250E.

Terminal Type

This field sets the type of 5250 terminal display you want to use. Here, the only available template is IBM-3812-1, which corresponds to a printer.

Host name

Enter the name of the server or its IP address. You can also select a server from the drop-down list (only enter a name if you have a names server).

Port

The default port number is 23. You can change this number, if necessary.

Unit name

Optional field. If you enter nothing in the field, the AS/400 system will attribute a default name to your client PC to uniquely identify it. You can, however, enter a terminal name to identify your machine.

Click on the Advanced button to customize the advanced properties of the 3812 session via a Netware for SAA or SNA Server gateway.

Identification

To enable automatic login, fill the required parameters when login to the AS/400 system: user name and password.

Select the Encrypt check box if you want the identification password to be crypted on login.

Queue Name

Specify the messages queue on the AS/400 (QSYSOPR by default).

Library Name

Specify the library on the AS/400 (*LIBL by default).

Font

Indicate here the IBM code of the font to be used. Refer to your IBM documentation for more details.

Terminal parameters

You can modify the default configuration of the AS/400 terminal on which the connection is established. Specify a new Code Page and/or Charset to use if necessary. Click Default to reset these values to your session defaults (given by the character set).

The UserVar Text and UserVar Binary lists let you configure other advanced parameters of the terminal. To add an additional IBM parameter (text or binary), click . Then, enter the name of the IBM parameter in the upper left part of the field, and its associated value in the right part.

For instance, you can enter the following additional parameter:



To remove a character from the list, select it and click .

Refer to your IBM documentation for more details on these settings.

Session tab (Netware for SAA or SNA Server connection on 3812 emulation)

This section describes the Session tab for 3812 emulation when the connection type is Netware for SAA or SNA Server.

Terminal type

Lets you choose the type of 5250 terminal. Here, the only available terminal is the IBM-3812-1 terminal, which corresponds to a printer.

System name

Enter the name or IP address of the server or select one from the drop-down list (only enter a name if you have a names server).

Unit name

Optional field. If you enter nothing in the field, the AS/400 system will attribute a default name to your client PC to uniquely identify it. You can, however, enter a terminal name to identify your machine.

Click on the Advanced button to customize the advanced properties of the 3812 session via a Netware for SAA or SNA Server gateway:

Mode name

By default, the mode name is QPCSUPP. This mode is specific to the gateway and APPC library. Typically, you don't need to change this value. Contact your network administrator if you want to change it.

Local LU name (only for SNA Server connections)

For connections through an SNA Server gateway, enter the name of the local LU as it's registered on the gateway. This field must be completed.

This field isn't shown for Netware for SAA gateway connections. The name of the local logical unit must be correctly registered on the gateway for the connection to succeed.

User name/Password

Enter the user name and associated password that allow you to connect to the AS/400 system. The user name and password are defined on the gateway.

Queue Name

Specify the messages queue on the AS/400 (QSYSOPR by default).

Library Name

Specify the library on the AS/400 (*LIBL by default).

3287 or 3812 Print Configuration

Print options for the server

These print options allow you to specify, for each printer emulation session, both mode and print format used by your MainFrame or AS/400 system server to send data to be printed to the PC.

Specify the 3287/3812 print options

Select the Terminal option of the Settings menu from the 3287 or 3812 print server.

In the Automatic print options, select the option for the automatic print mode that will be used by the server:

If the server sends an instruction indicating the end of the print job (EOJ), select Print on reception of End Of Job. When this instruction is received, the data is printed automatically.

If the server doesn't send this instruction, select Print after timeout. Once the print job is finished, the data will be automatically printed after the time in seconds specified in the Timeout field. If you don't specify a time in this field, the data will not be printed automatically. You must then start the print job manually

> See "Print Commands" on page 110.

The Rows section lets you define the width of the pages sent to be printed. Select the Default option to use the page format returned by the server: Alternatively, select Custom to specify a given page width or the number of characters printed per line.

The Columns section lets you define the height of the pages sent to be printed. Select the Default option to use the page format returned by the server: Alternatively, select Custom to specify a given page height or the number of characters printed per column.

PC Print Configuration

For each printer emulation session, it is necessary to configure the PC print mode and the printer used (if any) to print data coming from the IBM terminal.

Configure PC Print

Select the Print option of the Settings menu from the 3287 or 3812 print server.

Printing to a printer

Select the Use this specific printer option to print the received data on a printer connected to your PC.

Then, select the printer of your choice from the drop-down list. As in any other Windows application, the locally defined printers are available on it.

Choose Setup printer... to configure the printer on which you want to print.

Printing to a text file

Instead of directly sending print data received from the server, you can record the corresponding documents in one or more text file(s).

To do this, select the Print in a text file option.

Select the Print in a text file check box and deselect the Multiple Files check box to record the received documents one after the other in a single text file. Specify the path of the text file to be used or click on the Browse... button to select it.

To record documents in distinct text files, select Print in a text file and Multiple Files. Specify the path where the text files will be saved or click on the Browse... button to select it (the various file names will be created automatically from the directory name).

Orientation

Specify the print orientation for print jobs from the printer emulation session:Portrait or Landscape.

Non-graphic print

You can define the font used to print in text mode. You can set the font you choose as the default print font for all printings (including that requested from the Print option of the File menu) by selecting the Use selected font check box. Click Setup font... to specify the font to use and its style and size. Select Use size of selected font to set the font size as the default print size (if you don't select this check box, the next nearest size is used for printing).

Note:

The Script drop-down list box in the font definition dialog box lets you choose the character set encoding used by the font.

3287 or 3812 Emulation Status

You can query the status of the print jobs sent from the IBM mainframe server or the AS/400 at any time from the printer emulation session window.

This window gives you the connection and print status, and lists the following events:

- Connection and disconnection of the session.
- Print jobs.
- Communication with the server or printing errors.

Print session status

The upper-end of the window will provide you with information on the 3287 emulation status:



If no 3287 or 3812 emulation session is active on the PC, the 3287 or 3812 emulator status is Idle



If a 3287 emulation session is running on the PC and a connection is established with a host machine (IBM mainframe server), the 3287 emulator is Active.



If the printer is active (in the case of 3287) and an application on the IBM mainframe or the AS/400 system establishes a connection with the printer (via its LU), the 3287 or 3812 emulator switches to Session. It's then ready to receive and print data from the IBM machine application.



If a print job is being processed, the 3287 or 3812 emulator status is Printing.



If the print job is temporarily stopped with the Suspend command, the 3287 or 3812 emulator status is Suspended.

Print Commands

You can suspend and resume data sending; cancel printing or force the printing of received data.

Hold/Resume printing

Activate the print session window, and select the Commands ,then Hold printing or Resume printing options in the Settings menu. The 3287 or 3812 print session status is then changed to <Inactive>.

Cancel printing

When the print session status is <Inactive>, select the Commands and Cancel printing options in the Settings menu.

None of the received data is printed and the print job is considered as finished.

Force printing

When the print session status is <Inactive>, select the Commands and Flush printing options in the Settings menu.

The emulator will print the data already received. You must restart the print job for data to be received.

> See “Hold/Resume printing” on page 110.

Log file

You can record the information contained in the print emulation session window in a log file, that you may want to consult later on.

Specify the log file to be used

Select the Log... option in the Settings menu.

Select the Use a log file option to save the contents of the emulation session window in a log file (this option is not selected by default).

Enter both path and name of the file to be used (file extension .log) in the field, or click Browse... to select a file.

Hotspots

A hotspot is a screen zone identified by a character string. When the emulator recognizes this string in a screen, this leads to one or more controls being displayed on the screen (buttons, menus, texts, scrollbars, combined lists). Activation of the controls by the user (clicking on a button, choosing a menu option, clicking or double-clicking on a text, etc.) enables execution of one or more actions such as transmission of a character string or of a function key on the communication channel, performance of a script function, scrolling of a list, etc.

A hotspot is thus comparable to the notion of screen signature: detection of a given character string is significant of the status of the system and indicates the presence of a given screen in the program. This signature can be associated with one or more controls, which can be displayed in any zone on the screen, including over several lines.

Note:

One or more actions can be executed automatically when recognizing a string, without any control appearing and without any action on the part of the user. This type of “automatic control” is called “auto-active.”

Recognition and display rules

If several zones are recognized, the hotspot identified is that with the largest text. When several controls are liable to overlap each other, the only one displayed is the largest one.

If a control is to be displayed on the recognized string, the size of the control will be identical to that of the recognized string rather than that initially intended at creation of the control.

Types of controls

The different types of controls you can create for an active zone are as follows:

A button: one or more actions are performed when the user clicks on the button. The button text can be static or dynamic (updated by the emulator).

A menu: one or more actions are executed when selecting a menu option. The list of menu options appears when the button representing the menu is clicked.

Text: with neither button nor menu, you can trigger actions on a particular zone of the screen. Several cases are possible:

- A character string appears on the chosen screen zone, one or more actions are executed when you click on the text (single click, double-click, left, middle or right mouse button).

This is then a **Text** type control. The text displayed can be static or dynamic (updated by the emulator).

- If the control text is empty (no additional display on the screen), the control is then of the **Transparent text** type.
- As soon as the string to be recognized (definition of the hotspot) is identified on the screen, one or more actions are automatically executed with no intervention on the part of the user. For example: as soon as the "login" string is found on the screen, the emulator automatically sends the user1 string as the login. This is then an **Auto-active** type of control.

A drop-down list or combined list (combo box): the character string selected from the list is sent to the connection. The content of the list can be static or dynamic (updated by the emulator).

A scrollbar: one or more actions are executed when you click on a particular part of the scrollbar (page left, page right, left or right for a horizontal scrollbar and page up, page down, up and down for a vertical scrollbar). This enables you to move through the lists with the mouse.

Note on the combined lists:

In the case of asynchronous emulation, a Combined list type of control can only be active if the cursor is on its zone. The character string selected from the list is then transmitted after striking the Enter key on the keyboard.

In the case of synchronous emulation, several controls can be carried out simultaneously. When a control becomes active, the cursor is positioned in the zone it controls. The character string selected from the list is then executed when the cursor leaves the control.

Creating hotspots

You can configure hotspots in two different ways:

- For each string recognized on the application screens, you define a list of controls to be displayed (access as desired from menu or toolbar).
- You define a control directly for a string selected on the screen (access by context menu after selecting the string to be recognized).

List of controls per hotspot

All the following operations are performed using the Hotspots option on the Settings menu.

The list of hotspots takes the form of an objects tree:

- For each string to be recognized, there is a hotspot object.
- For each hotspot, there are one or more controls, characterized by their name, type and action to be executed at a particular event.

By default, the tree contains a single object, which is the root of all the others.

In this tree, you will configure the hotspot(s) of your choice, whose controls will be displayed on the screen when the associated character string is recognized.

Example:

When the login string appears, you want a combined list (combo box) to appear on the screen allowing selection of the user name from among those proposed and the “You have mail” string to be customized (translated into the users' current language and/or displayed against a colored background and with a particular font).

To do this, you must create a hotspot characterized by the character string to be recognized (login). You can then define two controls: one a combined list type which you will position next to the word "login" on the screen, the other a text type, which you will position on the “You have mail” string.

Creating a new hotspot








Go to the hotspots root and click on the New button.

Name

Enter the name you want to give to the hotspot created. This name enables the hotspot created to be identified in the hotspots tree.

Type

Select the type of the first control you want to associated with the hotspot created by pressing the appropriate button:

-
-  A button
 -  A menu
 -  Transparent text
 -  An auto-active control
 -  Text
 -  A scrollbar
 -  A combined list
-


> You can then add other controls to this hotspot. See “Creating a new control for a hotspot” on page 118.

String search

Enter the character string to be recognized. State whether:

- The case is sensitive (upper case recognized or not).

- The string represents a complete word. The check-box should be deselected if the character string may not be a complete word.
- The string is a regular expression containing generic characters (wildcards), such as *, ?.

If you select the Regular expression check box, you can choose wild-card characters from the list available, using the following button .

The wildcards available are as follows:

- . to search for any character.
- [] to enter a range in which the character looked for must be found.
- [^] to enter a range in which the character looked for must not be found.
- * to look for 0 to n occurrences of the character situated immediately to the left.
- + to look for at least one occurrence of the character situated immediately to the left.
- ? to look for 0 to 1 occurrences of the character situated immediately to the left.

All these wildcards can be looked for in the text if preceded by the character "\".

Search area

Select the Whole screen option to search for the string in the entire screen, or define a particular screen zone by selecting the Area option. In this case, you can:

- Either enter the coordinates of the zone directly in the boxes (in the order: row and column of top-left corner, row and column of bottom-right corner).
- Or manually capture the screen zone concerned by the search, after clicking on the Capture... button

Advanced

You can also specify the display attribute of the characters in the string to be recognized. To do this, click on the Advanced button.

Select the Use attributes check-box then select or deselect the attribute types you want to include in or exclude from the search..

If attribute A is deselected, the character string search will use all the attributes except attribute A.

If attribute A is selected, the character string search will only use type A attributes.

In the above example, the previously defined string will be identified by all attribute types, except the Underlined attribute.

If you are looking for a string containing a number of complete words, specify the characters which separate these words in the Split characters field. This field is available only if the Whole word option in the previous screen is selected.

Click on the OK button to return to the previous screen. Click on the Next button to move to the next screen.

General appearance of the control

In the Description section, enter a unique identification of the control.

In the Positioning section, select the option corresponding to the control position on the screen:

- On the hotspot: the control will position itself on the string to be recognized, defined when creating the hotspot.
- Fixed on screen: the control will position itself in the zone for which you enter the coordinates in the following order: row and column of top-left corner, row and column of bottom-right corner. As in the case of the hotspot, you can use the Capture... button and then capture the coordinates of the hotspot directly on the screen using the mouse.

Note:

In the case of an Auto-active type control, you do not define the position of the control, as an Auto-active type control triggers execution of one or more actions as soon as the hotspot is identified, without any particular display.

Click on the Next button to move to the next screen.

Appearance of button or text

You can state which text and/or image is to appear at the location of the chosen control.

To display the same text as the recognized zone, select the Use the identified zone text option. Otherwise, select the Display text option and enter the text you want to see displayed.

To display an image on the button, enter the absolute path of the image file chosen. The authorized formats are .bmp and .ico.

Note:

If, in order to modify the displayed text dynamically, you associate an action with the initialization phase of a Text type button or control, the dynamic text will take priority over the text entered here.

Defining the scrollbar

Select the orientation of the scrollbar (horizontal or vertical).

Select the type of scrollbar movement:

- With an indefinite number of steps (a step is made at each click on the end of the scrollbar): select the Undefined number of steps option.
- By setting scroll movement values: select the Fixed values option. Then enter the number of steps, the number of elements situated in a page for scrolling in page up/down mode (click inside the scrollbar) and the initial position of the scrollbar (in steps).

Actions to be executed

It can be used to customize:

- Button: the actions to be executed at initialization of the control and when clicking on the button.
- Menu: the actions to be executed at initialization of the control and when selecting each of the menu options.
- Text and Transparent text: the actions to be executed at initialization of the Text control and for each mouse event on the Text and Transparent text (single click, double-click, left, middle and right mouse buttons).
- Auto active: the actions to be executed as soon as the string defined for the hotspot is recognized.
- Scrollbar: the actions to be executed for each scrollbar movement (initialization after any click on the scrollbar, page left, page right, page up, page down, left, right, up, down).

Note:

Whatever the control, for each event in the broadest sense (initialization of control, button click, option selection, mouse event, string recognition, scrollbar movement), you can define several actions which will run as a sequence. Selecting the event in the left-hand window on the Actions screen calls up the list of corresponding actions in the right-hand window.

To allow execution of these actions only once, select the Disable after execution check-box.

- > To make up the list, refer to the explanations given in the “Modifying a control on a hotspot” on page 119 paragraph.

Text format

Note:

In the case of a Combined list type control, you will only define the font format and the effects.

In the Font section, select the name of the font you want to use for the text displayed:

- The default font proposed is System, and corresponds to the font of your client station. You cannot alter its size from the emulator.
- You can also select a particular font (for example Arial, Times New Roman, etc.) for which you can choose the size.
- Finally, you can choose the font displayed on the screen by the emulator (Emulator font). You cannot specifically change its size here.

In the Style section, select the check-box(es) corresponding to the style of your text: bold, italic, underlined.

In the Text section, select the option you require: text centered, justified left or right.

In the Image section, select the option you require: image centered, justified left or right.

You can select the control background color, the text color and the color which will be replaced by transparency in the image.

Creating a new control for a hotspot

In the list of hotspots (accessible from the Hotspots option in the Session menu), go to the mother hotspot and then click on the New button.

Select the type of control you want to create, then click on OK. Consult the detailed list of controls in the Type of control paragraph.

For each type of control, a particular wizard is started, helping you to customize the control. Follow the instructions given in each screen proposed by the wizard, referring to the explanations given previously.

> See “Creating a new hotspot” on page 114.

Creating a control from a zone selected on the screen

If you simply want to create a control for a string selected on the screen using the mouse:

- Display the context menu by clicking on the mouse right button after selecting the characters of the string to be recognized.
- Select the Create a hotspot option on the context menu.

The hotspot wizard then opens in the same way as when selecting the Hotspots option on the Session menu.

For each screen proposed, proceed in the same way as when creating a hotspot and its controls.

> See “Creating a new hotspot” on page 114 or “Creating a new control for a hotspot” on page 118.

Note:

After creating a control from a zone selected on the screen, the hotspot and its control created in this way appear in the list of hotspots defined in the session. This list is accessible from the Hotspots option on the Session menu. You can then add other controls to this hotspot.

Modifying and deleting a hotspot or control

Modifying a hotspot

After creation you can modify the definition of any hotspot: identification, string to recognize and search parameters.

Display the list of hotspots by selecting the Hotspots option from the Session menu.

Select the hotspot you want to modify from the list of hotspots, then click on the Modify button. Make the changes you want.

Deleting a hotspot or control

Display the list of hotspots by selecting the Hotspots option from the Session menu.

To delete a hotspot or a control from the list of hotspots, select the element and then click on the Delete button.

To delete all the elements on the list of hotspots, click on the Delete all button. Only the hotspots root will remain available for creating new hotspots and their controls.

Modifying a control on a hotspot

You can modify the definition parameters for a hotspot control in two ways:

- From the list of hotspots (Hotspots option of Session menu), select the control from the list then click on the Modify button.
- From the control selection on the screen: display the context menu by clicking on the mouse right button on the control displayed on the screen, then select the Hotspots properties... option

In both cases, a tab box appears (the number of tabs and their content varies according to the type of control). Modify the various parameters with reference to the Create a new hotspot paragraph.

Action editing tools on the controls

Events on a button

Two default events are proposed:

- Initialization, which corresponds to the moment at which the zone looked for is recognized and the control created. This phase is useful if you want to retrieve a character string and assign it as a button text.
- A click on the button.

No other event can be created.

Note:

Only the right-click event has to be associated with an action. Initialization is optional.

> See “Associating an action with an event or option” on page 121.

Menu options

A menu is in fact a button comprising several options. To associate actions with the menu, you must first of all define the options it is to contain, and then associate them with one or more actions.

> See “Adding an event (Text or Transparent text)” on page 121 or “Adding an option (Menu and Combined list)” on page 121.

> See “Associating an action with an event or option” on page 121.

Events on a text or transparent text

A text is a screen zone liable to react to mouse events. By default, a single event is proposed: initialization as in the case of a button (for dynamic text definition). You can then add the mouse events for which one or more actions are to be executed.

In the case of Transparent text type controls, this screen is similar but the text initialization option is not available.

- > See “Adding an event (Text or Transparent text)” on page 121 or See “Adding an option (Menu and Combined list)” on page 121.
- > See “Associating an action with an event or option” on page 121.

Events on an auto-active control

An Auto-active type control authorizes automatic triggering of one or more actions at recognition of a string. The only event linked to this control is recognition of this string.

- > See “Associating an action with an event or option” on page 121.

Events on a scrollbar

By default, the five events linked to a scrollbar are proposed:

- Initialization, executed before each other event. This phase in particular consists in repositioning the cursor in the zone controlled by the scrollbar, to enable execution of the actions associated with the events.
- Left/right and up/down movements.
- Page jumps (page left, page right, or page up, page down).

- > See “Associating an action with an event or option” on page 121.

Combined list options

The list is filled out when the string is recognized. Therefore it contains only the initialization event linked to the Combined list type control.

A list contains character strings. That chosen by the user will be sent on the communication channel. The list can be filled out in two ways: manually or by a script.

Manual method

You can yourself enter all the elements of the list in the editor proposed in this screen. To do this, select the List option (proposed by default) and with the editor add the elements as you want to see them appear in the list.

- > See Managing events or options below, for information on use of the list editor.

Script method

You can create a script which will be able to retrieve the information necessary for producing the list. The default language is EScript, Esker's proprietary language. Write your script in the script editor.

Managing events or options


In the case of Text and Transparent text type controls, you can add events to or delete them from the proposed list

In the case of Menu type controls, you must create the list of menu options with which one or more actions are to be associated.

In the other cases, the list of events cannot be modified.


In the case of Combined list type controls, you can create the list manually.

Adding an event (Text or Transparent text)

In the events list, click on the  button. You can also double-click on the empty field at the end of the list.

Select the type of event from the right-hand drop-down list and the combination key which can be used in the right-hand drop-down menu (option Normal for no key).


Adding an option (Menu and Combined list)

In the list of options, click on the  button. You can also double-click on the empty field at the end of the list. Enter the name of the option as you want it to appear in the menu or combined list.



Modifying the name of an option (Menu and Combined list)

In the list of options, double-click on the option for which you want to change the name, and enter its new name.

Deleting an event or option


In the list of events or options, select the event or option you want to delete and click on the  button or on the Del key on your keyboard.

Modifying the order of the options (Menu and Combined list)

In the list of options, use the  and  buttons to move a preselected option upwards or downwards.

Associating an action with an event or option

Select the event or option from the left-hand list.

In the list of actions, click on the  button. You can also double-click on the empty space at the end of the list.

Select the type of action to be executed.

String

In the field to the right of the type of action, enter the character string you want to send on the communication channel.

Function

In the drop-down list to the right of the type of action, select the function key you want to send on the communication channel.

In the case of an asynchronous emulation (UNIX type), the labels proposed are those listed in file `emul.fky`, copied into the emulators installation directory. The values assigned to the various function keys are identified in the function keys file available to each type of terminal emulated.

Note:

The function key files take extension `.fun`.

Mnemonic (only in asynchronous emulation)

In the drop-down list to the right of the type of action, select the mnemonic you want to send.

The mnemonics are short words interpreted by the emulator in order to perform a particular action.

The mnemonics of the Esker asynchronous emulator can symbolize:


- A single character (example: nul to send 0).
- An action (example: nlock to activate the Numlock key to lock the numerical keypad).
- A dead key, that is a key which only has an effect after another key is pressed (example: tilde to add a ~ to a letter).

> See “Customizing the keyboard” on page 86.

Macro


Note:

This feature enables you to use the macro files created with the macro language in the previous versions of Tun (Esker proprietary language). The macro files are text files with extension `.mac`.

In the edit field to the right of the type of action, enter the path for the macro file (`.mac`) or select it by clicking on the  button.

Script

- This option allows to enter several script lines directly, without any reference to any existing script file. The proposed language is EScript, Esker's proprietary language.

Click on the  button in the right edit field to start the script editor.

Function-Key Panel Editor

A function-key panel is a window containing buttons that you can summon during an emulation session. The following operations can be executed from panels:

- Transmission of a pre-defined string of characters on the communication channel (simulates a keyboard entry).
- Execution of a macro.
- Execution of a specific action (launch an application, make a hard copy...).

The key panel buttons may contain:

- Text (centered, left or right justified, on several lines).
- An image (centered, or tiled).
- An image and text.

Just like the keys on the keyboard that are changed by the Shift or AltGr keys, the buttons on a key panel can have three levels. Each button level can display a different value and carry out a specific operation. The level is defined by a special button called the Lock button.

The purpose of the key panel is to allow you to replace the keyboard with the mouse as much as possible. With well-designed key panels, traditional operations can offer the same ease of use as native Windows applications.

The aim of the present chapter is to explain how to design and define a function-key panel.

Starting the function-key panel editor

You can start the function-key panel editor from the **Start** menu in the same way as any other program, or from an emulation window using the Function-key panel Editor option on the Tools menu.

> Creating a panel

Select the New option on the Options menu.

Creating a button

There are two types of buttons:

Simple buttons that control three levels (just like the keys of an ordinary keyboard).

Lock buttons that change the state of all the simple buttons on the same key panel.

There can only be one Lock button per key panel.

> **Creating a simple button**

To create a simple button, do the following:

Click the toolbar button .

Click inside the key panel definition window and, to draw the size and shape of the button you want, slightly move the mouse with the button held down. Release the mouse button. This creates a blank button on the screen.

You now have to link properties to the button. To do this, double-click the blank button to display the following dialog box (you can also use the button context menu and select the Button parameters option).

The dialog box for the definition of simple buttons has three levels indicated by the Level 1, Level 2 and Level 3 tabs. Each level represents one of the three possible states the button can have. The same information is entered at each level.

Text

Enter the text you want to see on the button in the Text field.

Tooltip

Enter the tooltip text. This text appears when you place the mouse pointer on the function-key panel button.

Image

You can include an image of the paintbrush type (file extension .bmp) on the button by entering the full path name in the Image field. To select the image, use the Browse... button.

Settings

Click one of the Settings buttons if you want to specify options for the text and/or image.

The Text Information tab lets you add settings for the button text:

- Justification.
- Characters per line (for justified text).
- Horizontal and vertical margins in pixels.
- Text position: Above, on top of, or below the image.
- Horizontal and vertical text position.

The Font... button allows you to choose the font, character style, size, color, and effect.

Use the Bitmap Information, tab to set options for the button image:

Select the Color transparency check box to make one of the colors on the button transparent. By default, the color white is transparent (if you select the check box). You change this by clicking the Color button. The color chosen appears on the Color button.

Select the image position: Centered or tiled.

Click the Default button to assign the default values. If you want to apply your settings without quitting the dialog box, click Apply. Click Save to save your settings and close the dialog box, or Cancel just to close the dialog box without validating your settings.

Action

The Action field contains the string of characters that's emitted or the action taken when you click the panel key in an emulation session.


These actions and strings are given line by line according to the order of execution.

Click the button  in the Actions section to add a new action.


The default type Text is proposed (send a string of characters) but you can choose a different type from the list:



- Function-key (assigns a function-key to the button)
- Macro (executes a macro).
- Action (executes an action listed in the Escript.doc file).
- Other (for function-key panels created with earlier versions of Tun than 8.50).

Give details of the action in the right field. This depends on the action you selected:

- Text: Enter the character string to send.
- Function-key: Enter the function key number, or select it from the list that is displayed.
- Macro: Enter the macro path (.mac file) or use the button  to select the macro.
- Action: Enter the action number or its name. See the Escript.hlp file for the available actions (this help file is available in the \Docs\Misc directory of the CD-ROM).


To change an action you've already defined, double-click it in the list of actions.

To delete an action, select it in the action list and click the button  in the Actions section.

When a button is associated with more than one action, you can decide the order in which they're executed. To do that, use the button  to move the action up the list and  to move it down the list.

> Creating a Lock button

A Lock button in a function-key panel means that the number of buttons present in a function-key panel can be divided by three, by multiplying the button combinations. There can only be one Lock button per function-key panel.

To create a Lock button, carry out the same operations as for an ordinary button but first of all click the toolbar  button.

Just as in the case of an ordinary button, you can associate properties with the lock button by clicking it. The dialog box displayed is the same as for a simple button except that no Action is required. The sole purpose of the Lock button is to change the state of the other buttons on the function-key panel.

The Lock button works in the following circumstances:

- When you place a text or an image on the lock button at level 3, the button has three different states.
- When you place a text or an image on the lock button at level 2 and there is nothing at level 3, it can only have two states.
- If you place a text or an image on the lock button and there is nothing at levels 2 and 3, it can only have one state, which isn't very useful.

Additional buttons

One key panel can contain up to 128 buttons.

Default Button Settings

You can define default button settings which are used automatically for any new button.

Choose Default Button Settings... from the Options menu. If the option isn't available (the option Selected Button Options appears instead) this means a button in the model key panel is selected. Make sure no buttons are selected before choosing the option.

A dialog box similar to the Selected Button Settings dialog box opens.

> See "Creating a simple button" on page 124.

Panel Settings and Positioning

You can define the settings assigned to the current function-key panel.

Choose Panel Settings from the Options menu or select the option Panel Settings from the context menu. To display this menu, click the panel with the right mouse button.

Panel type

Select the type of emulation you're going to use (3270, 5250 or asynchronous emulation).

Title bar

Select the type of panel title bar you want (normal, half-height or none).

Panel name

Enter the name of the key panel for each button level. Click the arrow beside the field and choose the level (level 1, 2 or 3).

Docking


Select the type of docking you want to give the function-key panel:

- None: The function-key panel can't be transformed into a toolbar.
- Normal: The function-key panel becomes a normal toolbar.
- Wrapped: If there are too many keys, they're re-organized on more than one line or column.
- Don't change: The function-key panel changes into a toolbar, but the layout of the keys remains the same, irrespective of the screen configuration.
- Panel paste: The function-key panel is positioned in the emulation window as defined by the Possible positions fields (centered, top, bottom, left or right). The panel doesn't change into a toolbar.

If you choose Normal, Wrapped or Don't change, select the possible positions for the key panel (top or bottom, right or left).


The type of docking selected here is the Default option proposed by the emulator when the function-key panel is selected. See the chapter on defining the display settings for an asynchronous emulation session context.

If you want to use Panedit's default values, click Default. Click Apply to apply the settings without closing the dialog box. Click Save to validate your settings and close the dialog box, or Cancel to close the dialog box without validating your choices.


To view your changes, choose Test Mode from the Options menu, or click the  button in the toolbar.

Set Tab Order

The tab order controls the numerical order in which the buttons are selected or highlighted when the user presses the Tab key.

The Tab order of the buttons can be set by choosing Set Tab Order from the Options menu or clicking on the  button on the toolbar to display the order of the function-key panel buttons and change it if necessary.

To assign tab order to a panel of buttons, choose Set Tab Order from the Options menu and click the panel button which is to be number one with the right mouse button. Then click the remaining buttons with the left mouse button in the desired tab order. The order can be changed by restarting the operation.


When you're satisfied with the order, clear the menu option (or click the  button again in the toolbar).

Note:

Lock buttons always take the same number as the number of buttons.


Opening an existing function-key panel

You can adapt an existing function-key panel. Examples are supplied by Esker: They can be found in the emulators' installation directory. A panel file has the extension `.pan`.


To open an existing function-key panel, click the button  in the toolbar or choose Open from the File menu.


Choose the key panel file you want to open.

Saving a function-key panel

You can save your function-key panel in a file with the extension `.pan`. To do that, click the button  in the toolbar, or choose Save from the File menu (or Save As... to save the function-key panel under a different name). Enter the name of the function-key panel, if necessary.

Testing a key panel


To effect an immediate test of the key panel without having to load it into the terminal emulator, click the toolbar button  or choose Test Mode from the Options menu. This function simulates the use of the key panel, just as if it were being used in the terminal emulator.

To stop the test, click the same button again .

Supplementary functions of the panel editor

> Selecting one or more buttons

Click a button to select it.

To select more than one button at the same time, click the button  in the toolbar, and do one of the following:

Draw a rectangle around the buttons you want to select.

Or hold the Shift key down, and click the buttons you want to select.

> Moving and changing the size of a button



To move a button or change its size, select the button with the mouse. This displays dimensioning handles at each corner. Click the inside of the button and move the mouse with the mouse button held down to move the panel button correspondingly.

You can select and move several buttons at the same time.

> **Duplicating a button**

To duplicate a button, select it and do one of the following:

Hold the Ctrl key on the keyboard down and move the mouse. When you release the mouse button, an identical button to the selected button appears.

Or click the button , and then the button  in the keyboard.


Or choose Copy and then Paste from the Edit menu.

All the data on the button, text or bitmap, is copied.

> **Deleting a button**

To delete a button, select it and do one of the following:

Use the Del key on the keyboard.

Or click the button  in the toolbar.

Or choose Cut from the Edit menu.

In the last two cases, the button is copied to the clipboard. You can recover the button by using the **Paste** function (button or menu option).

> **Aligning the buttons**

You can align and redimension buttons as follows:

- Align the buttons on a grid.
- Align the buttons in relation to each other.
- Center the buttons in relation to the panel (vertically, horizontally).
- Apply the same dimensions to several buttons (same width, same height or same dimensions).

Aligning the buttons on a grid


By default, the buttons are positioned in the panel as you draw them.

To change the alignment grid settings, choose Align from the Options menu.

To activate the grid, select Use grid. Select Display grid to display the grid.

You can also enter the spacing between the horizontal and vertical lines of the grid in pixels. The default value is 10.

Click OK.


To activate the grid, you can also click the button  in the toolbar. The grid settings (display, number of pixels) are those defined in the Alignment dialog box. If the settings aren't

suitable (the grid isn't visible, or the spacing is too large, etc.), choose Align... from the Options menu and change the settings.


Relative button alignment

To align several buttons (at least 2) in relation to each other, select them, and then click the appropriate button in the toolbar:



 Aligns buttons on the left side.

 Aligns buttons on the right side.

 Aligns buttons along the top.

 Aligns buttons along the bottom.

Centering buttons

To center a button in the key panel, select it and click the button  in the toolbar to center it vertically and/or the button  to center it horizontally.

Attributing the same dimensions to several buttons

To give several buttons the same dimensions, select them and click the appropriate button in the toolbar.

 Same width.

 Same height.

 Same height and width.

> Changing the description of a button quickly

To change the descriptions and actions of a button, you can update the table at the top of the panel editor screen.

Context menus and toolbar

Most of the above options are accessible through context menus.

Panel Settings




Click an empty space in the prototype panel with the right mouse button to access the Panel Settings dialog box.

Button Settings

To display the button menu, first select the button by clicking it with the left mouse button. Then click once with the right mouse button.

Toolbar

The button functions are described below:

| | |
|---|--|
|  | Opens a new panel file and closes the current panel file if one is open. |
|  | Opens an existing panel file. |
|  | Saves the panel file. The default extension is .pan. |
|  | Cuts the selected button(s) and copies it/them to the clipboard. |
|  | Copies the selected button(s) to the clipboard. |
|  | Copies the button in the clipboard to the current panel. |
|  | Normal selector arrow. Can be used to select one or more buttons in the model panel. |
|  | Button arrow. Click to draw a new button. |
|  | Lock button arrow. Click to draw a lock button. Only one lock button is allowed per panel. |
|  | Toggles test mode on and off. |
|  | Sets tab order. Click the first button with the right mouse button and number the remaining buttons by clicking them with the left button. |
|  | Aligns the selected buttons on the left side. |
|  | Aligns the selected buttons on the right side. |
|  | Aligns the selected buttons along the top. |
|  | Aligns the selected buttons along the bottom. |
|  | Centers the selected buttons vertically. |
|  | Centers the selected buttons horizontally. |
|  | Attributes the same width to the selected buttons. |
|  | Attributes the same height to the selected buttons. |
|  | Attributes the same width and height to the selected buttons. |
|  | Activates alignment. |

Emulator Automation

This chapter is intended for service providers, system integrators, and information services staff who want to provide their users with transparent access to UNIX resources.

The techniques explained in this section let you completely automate connection, disconnection, file transfer, UNIX program startup, and other actions.

Macro principles

Esker's emulators include a high-level script-language (EScript) which serves to create scripts that replace the keyboard and totally or temporarily control an emulation session. You can call such a script when you start the emulator, and/or when you exit emulation.

For example, you can use EMUL's script-language instructions to do the following:

- Send a character string over the connection.
- Wait for a particular character string from UNIX within a specified time.
- Wait for a period of time expressed in seconds.
- Retries.
- Test return codes of certain instructions.
- Test characters received.
- End an emulation session.
- Return to an emulation session.
- Show or hide characters sent by the host.
- Prompt the user to enter information.
- Handle variables.

Syntax

You can create macros with an ordinary text editor. You don't have to compile them since they're directly interpreted by the emulators.

One of the most common ways to use macros is to associate with the session start and end macros, used respectively when you connect and disconnect the session.

Sample macro

The following script example automates logon to a server and launches a UNIX application (scoadmin).

```
Module scoadmin

# Variable used to store the number of the matched string:
Dim Matched as Int

# Characters sent by Host computer not displayed:
Dialog.SetTitle("SCOADMIN")
SetDisplayOff()

# Start:
Label BEGIN

# Read login and password:
Dim USER as String
Dim PASSWD as String
USER = Dialog.Read("Enter your user name: ")
PASSWD = Dialog.ReadPasswd("Enter your password: ")

# Make connection:
Repeat 3
    # Send carriage-return character:
    Repeat 5
        SendString("\n")
        If Receive(1000,"ogin") > 0 Then Break Endif
    EndRepeat
    If Error() Then Goto NOCONNECTION Endif

# Send login:
SendString(USER + "\n")
Matched = Receive(15000,"assword","# ","$ ")
If Error() Then continue Endif
If Matched >= 2 Then Break Endif    # "#" or "$" matched
SendString(PASSWD + "\n")
Receive(15000, "$ ", "# ", "ogin:", "TERM =")
If Error() Then Continue Endif
If StringFound() = "# " Then Break Endif
If StringFound() = "$ " Then Break Endif

# Return to start of program if login incorrect:
If StringFound() = "ogin:" Then Goto BEGIN Endif

# Set the TERM variable if necessary:
SendString("\n")
If Receive(15000,"# ", "$ ")>0
    Then Break
    Else Continue
Endif
```

```

EndRepeat

# Start application:
SendString("scoadmin\n")
# Display received characters:
SetDisplay()
# Return to the emulator:
Return

# No login:
Label NOCONNECTION
Dim ANSWER
ANSWER = MsgBox("Communication failed\nDo you want to quit Emul?","SCOAD-
MIN",4)
If ANSWER = 6
    Then exit          # Exit the emulator
Endif
CloseSession

```

The same type of program can be designed to establish more complex connections, such as sending modem commands, connecting through an X.25PAD, navigating within a Unix application, etc.

The preceding model can also be used to run UNIX applications other than scoadmin. For example, you can replace the line `SendString("scoadmin\n")` by a `SendString` method using another UNIX command or shell script as parameter.

Language syntax

An EScript macro must always begin with the word `module` followed by the macro's name.

The macro can then contain the following elements:

- Control structures. loops with stop and continue conditions, conditional blocked instructions, jumps from one place in a script to another.
- Variables of the following types: integer (`Int`) or character string (`String`).
- Macro parameters (used when you specify start and end macros from the emulator).
- Comparison, arithmetical and boolean operators.
- Predefined functions and methods.

Notes:

In this manual, instruction names are combinations of upper- and lowercase letters: This is only to simplify reading. It's of no importance with respect to programming, and either upper- or lowercase letters can be used to write instructions (for example, *SendAndReceive* can be written *SEND-ANDRECEIV* or *sendandreceive*).

If the first character of a line is `#`, the entire line is treated as a comment.

For more details about this script language syntax and functions, refer to the file `Escript.hlp`.

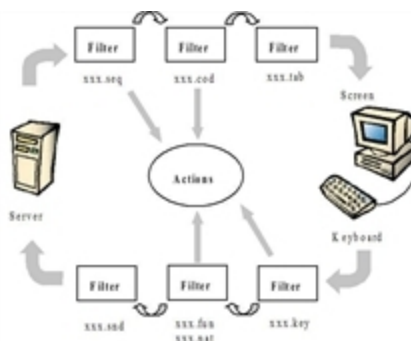
Advanced Use of Asynchronous Emulator

The concept of emulation

Given the wide variety of requirements for terminal emulators, Esker's asynchronous emulator was designed to let you define and customize every aspect of an emulation session, including keyboards, escape sequences, and character tables.

The emulation parameters are grouped together in an entity called the terminal. Each type of terminal (file .ter) is associated with various files containing the information necessary for communications between the PC and the server: keyboard file .key, function keys file .fun, escape sequences file .seq,... By default, Esker proposes dozens of different types of emulations, which you can customize to your own specific requirements.

The following diagram shows how the asynchronous emulator handles emulation:



The emulation cycle above shows how the data flow passes through different filters. The filters perform different operations on the data depending on the type of data and the filter settings.

Data from the PC to the server goes through the following filters:

Keyboard filter (for .key files) : each code sent from the keyboard (each key is identified by a code known as the scan code) refers to a piece of information. This information can be:

- A character (or a string of characters) to send.
- A script or a macro of type .mac to be executed.
- A function key (link with the function keys file .fun).
- A mnemonic. If the mnemonic is "nat", this means that you should refer to a specific national feature (national file .nat).

- Function-key filter (.fun file): Each function-key is assigned a value.
- Code conversion filter (.snd file): Converts ASCII files in some types of emulation.
- National filter (file .nat): the scan code sent by the keyboard can be re-routed by the .key file to a value of the .nat file specific to the language used.

In the next stage, the data reaches the UNIX host. The host processes the data and sends a reply to the PC. The reply is also filtered:

Escape sequence filter (for .seq files): The emulator processes the escape sequences sent by the UNIX host and links them to one or more actions (clear the screen, move the cursor, start an application, etc.).

Control code filter (for .cod files): When the host sends special characters known as control codes (the decimal values 0 through 31 and 128 through 159), Tun EMUL looks up the actions mapped to the characters in a table).

Character table filter (for .tab files): Displays the characters correctly on the screen. It ensures UNIX host codes are correctly interpreted by the emulator.).

You can modify the .key, .fun and .seq files for particular sessions. You seldom need to modify the .nat, .snd, .cod and .tab files.

All the configuration files are text files which can be opened in a Notepad-type utility and then modified.

You can however access these files and configure them using the Terminal option under the Settings menu on the asynchronous emulator.

If you chose a terminal type for the current session, the corresponding filename appears in the title of the dialog box (here ansi.ter). All the files displayed correspond to the parameters associated with this type of terminal (for example: keyboard file ansi.key, function keys file ansi.fun).

If you want to load another type of terminal, click on the Load button and select the .ter extension file of your choice.

The files referred to in the various fields can be edited from this configuration box by selecting them and then clicking on the Modify button. For a .key file (Keyboard field), a model of the keyboard appears. For all the other file types, the file itself is opened in Windows Notepad.

- > To understand how to modify the keyboard definition, see the **Using the emulators** chapter, **Customizing the keyboard/Asynchronous emulation** section.

Escape sequences

The asynchronous emulator uses .seq files to interpret the flow of data from the host machine. The .seq files associate one or more actions (cursor movement, clear screen...) with the reception of character strings (typically called escape sequences).

> **Opening an existing .seq file**

Out of an emulation session

A .seq file is a text file that can be read by a Notepad-type text editor. You can therefore open a .seq file from an editor of this type. The escape sequence files supplied by Esker are copied into the emulator's installation directory.

During an emulation session

Select the Terminal option from the Settings menu.

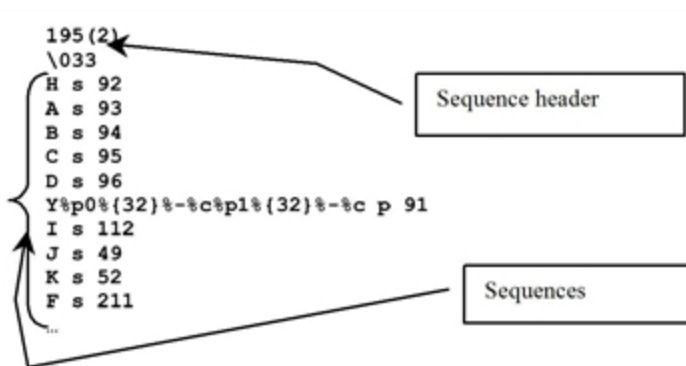
Click on the .seq file chosen from the drop-down menu of Escape sequences then click on the Modify button. The .seq file opens in the default test editor.

Content of an escape sequences file

A .ses escape sequences file comprises three separate parts:

- Terminal initialization, which enables the terminal to be set to the initial status needed for establishing communication between the server and the terminal. This part can be described in the first line(s) of the file.
- The escape sequences header, when common to all the sequences (optional part).
- Definition of the escape sequences.

Here, for example, is an extract from file vt52.seq:



Escape sequence files may need editing in the following cases:

- If the current initialization strings are not appropriate.
- If the action associated with an escape sequence is not appropriate.
- If you want to add escape sequences for particular actions.

Syntax

The escape sequences and the initialization strings are described by actions specific to the Esker asynchronous emulator. Each action is identified by its number or label and its parameters, if any. The list of asynchronous emulator actions is given in the Escript.hlp file, copied into the installation directory.

Note:

If an emulation session has been customized or newly defined, the File Reception feature (with

<Alt><F8> and <Alt><F9>) will help you to capture and analyze escape sequences and display characters sent by the host (using a Debug utility).

Terminal Initialization

The first line of an escape sequence file contains the list of actions necessary for the terminal to function properly. You can add or replace actions according to your needs (see following paragraphs for information on obtaining on-line help with escape sequences).

The Initialization line contains different actions separated by spaces. Parametered actions must be written with the appropriate parameters enclosed in parentheses and separated by commas. If there are a lot of actions, you can divide the initialization sequence into several lines, terminating each line, except the last one, with the backslash character \ (for example, the 2nd line in the file wyse60.seq).

Here, for example, is an initialization line:

```
195(0) 1195(0) 196(2) 197(2) 216
```

or with the actions label:

```
TabAsG1(0) TabAsG2(2) TabAsG3(2) G2IntoGR
```

These actions are defined as follows:

| Action | Description |
|--------|---------------------------------------|
| 195(0) | Assignment of character table 0 as G1 |
| 196(2) | Assignment of character table 2 as G2 |
| 197(2) | Assignment of character table 2 as G3 |
| 216 | Lock G2 in GR |

Sequence Headers

If all the sequences in an emulation begin with the same characters, it's best to enter them on the second line of the .seq file. This line serves as a header for every line that follows, and allows the emulator to treat sequences sent by the server more rapidly. The escape character (\033) is very often used a header.

If you don't use a Sequence Header, you must leave the second line blank.

Defining escape sequences

The remaining fields define the actions that are related to a particular sequence. There are two types of sequences:

Simple sequences that don't change.

Parametered sequences that may vary.

Simple sequences

A simple sequence is a character string that doesn't contain a variable zone, and may be directly associated with one or more actions.

For example, here's a string of three characters that move the cursor one position to the left:

```
\E[D s 96
```

or with the actions label:

```
\E[D s MoveCursorLeft
```

Parametered sequences

A parametered sequence is composed of a succession of strings beginning with the character %, which serves to identify the presence of a variable. A sequence can contain several parameters, defined in three parts:

Definition of the parameters themselves.

Calculations and controls to be carried out on the parameter.

Parameter format.

Parametered actions

In the case of a parametered action, there are two possibilities:

- The escape sequence is simple: the action parameters are constants.

Example:

```
c s 270("vt100")
```

(or c s ChangeTerminal("vt100"))

Escape sequence `\033c` executes the 270 ChangeTerminal action (dynamic terminal change) whose parameter value is vt100.

- The escape sequence is parametered: the parameters are in the order expected by the action, which then recovers the values taken from the sequence.

Example:

```
Y%p0%{32}%-c%p1%{32}%-c p 91
```

(or Y%p0%{32}%-c%p1%{32}%-c p MoveCursor)

Escape sequence `\033Y` executes the 91 MoveCursor action by recovering the values of two parameters (p0 for columns and p1 for rows).

In addition, after the action retrieves the parameter value, operations can be performed on this parameter before it is used by the action.

Example:

```
31 (-30) [30, 37]
```

- Check whether parameter value is between 30 and 37. If not, the action will not be carried out.
- Subtract 30 from the parameter value before it is used by the action.

Defining parameters

Note:

In the following notations, [] indicates an optional range.

Parameter definition uses the following syntax:

- %[?value by default]p[0-9] Allocation of a parameter
- Example %?1p2 Third parameter with a default value = 1
- %[?value by default]pi Allocation of several parameters
- Example %?1pi
- %g[a-z] Allocation of a variable
- Example %gh Allocation of the variable h

Calculations and controls

In conventional mathematic notation, the operands are separated by the operators and a particular significance forced by using parentheses. In RPN, the operands and operators are simply stacked and “popped” and “pushed” accordingly.

| Operator | Function | Examples |
|------------|---|---|
| %[min,max] | Checks the contents in the range | %[0x40,0x7f] The variable must be between 0x40 and 0x7f |
| %'c' | Stacks a constant | %'b' |
| %"string" | Push the string <i>string</i> on the string stack | %"green" |
| %{nn} | Stacks a decimal constant | %{64} |
| %g[a-z] | Pops a variable off the stack | %gh |
| %P[a-z] | Stacks a variable | %Ph |
| %V | Push the vertical position of the cursor on to the integer stack | |
| %H | Push the horizontal position of the cursor onto the integer stack | |
| %+ | Add | |
| %- | Subtract | |

| Operator | Function | Examples |
|----------|----------------------|--------------------------------|
| /* | Multiply | |
| / | Divide | |
| %m | Modulo (remainder) | |
| %& | Bitwise AND | |
| % | Bitwise OR | |
| %^ | Bitwise Exclusive OR | |
| %= | Equality | |
| %> | Greater than | |
| %< | Less than | |
| %A | Logical AND | |
| %O | Logical OR | |
| %! | Logical NOT | |
| %~ | Bitwise NOT | |
| %I | Bitwise Inverse | (01100010 becomes 01000110) |

Parameter format

This is shown as follows:

| | |
|--|--|
| %c | Single character |
| %s | Character string delimited by " or " |
| %S(string) | Character string ending by string. string isn't stacked, and must be less than 10 characters. The decimal, hexadecimal and octal notations must begin with the character \. The character) mustn't be used within string, and must be coded \0x29. Note: %S() represents a character string ended by the first received character. |
| <pre>% [[:]flag] [dim[.precision]][type]</pre> | |
| flag | Can have the values - + or # |
| - | The result is centered to the left. |
| + | The result always includes the sign + or - |
| Blank | If the first character of a conversion with a sign does not have a sign, a space precedes the result. This implies that if the flags blank and + are listed, the blank flag isn't taken into account. |
| # | This flag means that the value has to be converted to a format depending upon the type of the corresponding argument. This flag has no effect on the |

| | |
|-----------|---|
| | type d. In the case of a conversion of type o, it raises the precision in such a way as to force the first digit of the result to 0. In the case of a conversion of type x or X, a result other than zero is prefixed as 0x or 0X |
| dim | Gives the minimum number of characters occurring in the parameter. If this dimension begins with '0', the number is padded on the left by zeros and not blanks. |
| precision | Indicates the required number of digits (and not characters) corresponding to the parameter. |
| type | Can have the following values: d, o, x or X. |
| d | A signed decimal is converted into a integer |
| o | An octal notation is converted into an integer |
| x | A non-signed hexadecimal is converted into an integer (use the lowercase letters a, b, c, d, e and f). |
| X | A non-signed hexadecimal is converted into an integer (use the uppercase letters A, B, C, D, E and F). |

Examples

Setting the mouse for ansi emulation

```
\033Mm%p0%d; %p1%dX
```

There are two parameters in this sequence:

- %p0%d : first parameter
- %p1%dX : second parameter

These parameters are a succession of digits indicating an integer (d).

Reassignment of a keyboard key for ansi emulation

```
\033Q%p0% [0, 9] % { 59 } %+%d%p1%S ()
```

There are two parameters in this sequence:

- %p0% [0, 9] % { 59 } %+%d : first parameter
- %p1%S () : second parameter

The first parameter is an integer expressed in decimal format, while the second is a character string bounded by the first character received.

For the first parameter, the following processing is required:

| | |
|--------|---|
| %[0,9] | Controls that the character is in the range of decimal values 0 through 9 |
| %{64} | Stacks the value 59 |
| %+ | Addition in Reverse Polish Notation: (car59+) is equivalent to (car+59) |

The function keys

The emulator uses the .fun files to define each of the function keys used by an emulation.

> Opening an existing .fun file

Out of an emulation session

A .fun file is a text file that can be read by a Notepad-type text editor. You can therefore open a .fun file from this type of editor. The function key files supplied by Esker are copied into the emulator's installation directory.

During an emulation session

Select the Terminal option from the Settings menu.

Click on the .fun file chosen from the Function keys drop-down menu and then click on the Modify button. The .fun file opens in the default text editor.

Content of a function keys file

A **.fun** function keys file associates each function key on the terminal with the character string to be sent when it is struck.

Here, for example, is an extract from file vt100.fun:

```
[fKeyActions]
fKey1=\033OP
fKey2=\033OQ
fKey3=\033OR
fKey4=\033OS
fKey5=brk
fKey6=\033[17~
fKey7=\033[18~
fKey8=\033[19~
fKey9=\033[20~
fKey10=\033[21~
...
```

If necessary, you can change the value associated with each of the function keys. Character string encoding follows the same rules as those defined for the character strings of the .key keyboard files.

Integration of function keys in the emulator

The function keys can be invoked when striking a key on the keyboard or at a mouse event. Reference to the function keys file can thus be made in the keyboard and mouse definitions.

The `emul.fky` file is used to display simple labels for the various terminal function keys in the keyboard and mouse configuration boxes.

Example

Function key `fKey22` has label Shift F10: when a keyboard key or mouse event is associated with function key Shift F10, the keystroke or mouse activation corresponds to function key `fKey22` whose value is defined in the function keys file (for example `fKey22=\033[34~` in vt100 emulation).

Terminal configuration

You can associate a terminal configuration file (extension `.ses`), whose contents interact with the `.seq` and `.cod` files, with each type of terminal. A number of configuration files are supplied with Esker's emulator so that standard configuration parameters for the chosen terminal can be associated with each session.

> Opening an existing `.ses` file out of an emulation session

A `.ses` file is a text file that can be read in a Notepad-type text editor. You can therefore open a `.ses` file from this type of editor. The terminal configuration files supplied by Esker are copied into the emulator's installation directory.

> Opening an existing `.ses` file during an emulation session

Select the Terminal option from the Settings menu.

Click on the `.ses` file chosen from the Parameters drop-down menu, then click on the Modify button. The `.ses` file opens in the default text editor.

Content of a terminal configuration file

Here, for example, is an extract from the `vt220.ses` file:

```

[Intro]
ID=19971009
ParamNb=5
Param1=Cursor
Param2=Wrap
Param3=KeyMode
Param4=Keyboard
Param5=AbortEsc

[Cursor]
Label=SetupCursorStyle
ItemNb=2
Item1=SetupCursUnderline
Item2=SetupCursBlock
Action1=127(12,14)
Action2=127(0,14)
InitDefault=1
InitAction=%gS136%(16)%/(5)%>%(2)%{(1)%@

[Wrap]
Label=SetupAutowrap
ItemNb=2
Item1=SetupON
Item2=SetupOFF
Action1=62
Action2=63
InitDefault=1
InitAction=%gS4%(2)%{(1)%@
...

```

A .ses file is used to define the different terminal configuration parameters (for example, cursor style, keyboard type, sequence abort or not, etc.). Each section of the .ses file describes the various possible options for a parameter which can be selected by the user in the terminal configuration box.

Details

Identifiers

The names of combo-boxes and the listed items are stored in the emulator's language files (.lg). The file .ses contains the related name file identifiers (SetupCursorStyle, SetupAutoWrap, etc. in the example) as parameters.

Order of the combo-boxes

The combo-boxes appear in the order of their definition in the section [Intro]. The items listed in a combo-box appear in the order of their definition.

Actions

The field InitAction must contain a number from 1 through N. This number is the initial choice in the combo-box when it's opened in an active session. In other cases, the field InitDefault is used. InitDefault is set in accordance with the initialization parameters of the .seq file. That way, the .ses and .seq files aren't contradictory.

If the dialog box relates to the active session, the actions linked to the options in the combo-box (Action1...ActionN) are executed when the dialog box is validated (OK is pressed).

Loading order of the .ses file

The .ses file is loaded after the .seq file but before the configuration file containing the users' choices.

The national keyboards

In highly-specific multi-lingual environment situations, the emulator enables national filters to be employed, for example so that particular accents can be used. Using the nat mnemonic, it is then possible to assign a particular value to a keyboard key, as referenced in the .nat file. The values referenced in a .nat file are simple characters or mnemonics.

Example:

You are working in a French environment (azerty keyboard). However, a user temporarily wants to use a US keyboard configuration (qwerty keyboard). After selecting the US keyboard from the National keyboard option, the user can then redefine the keyboard by associating the nat mnemonic with key «a» on the PC keyboard. This indicates that the emulator has to refer to the .nat file to find out the value of this key in a qwerty environment (in fact, striking the «a» key will display a «q» on the screen, as if a qwerty keyboard were being used).

Reading a .nat file

When a PC keyboard key refers to the nat mnemonic, the character or mnemonic associated with it is located on the row corresponding to the key's scan code. The first column of the row gives the scan code, while the eight other columns give the value assigned to the key, in the following order:

- Key without combination (basic)
- Shift
- Ctrl
- Ctrl/Shift
- Alt
- Alt Shift
- Alt Ctrl
- Alt Ctrl Shift

Example:

The keyboard key with scan code 16 refers to row 16 of the .nat file. The value of this key, when combined with the Shift key, is defined in the third column of row 16.

> **Opening an existing .nat file**

A .nat file is a text file that can be read in a Notepad-type text editor. You can therefore open a .nat file from this type of editor. The national files supplied by Esker are copied into the emulator's installation directory.

Control codes

Characters whose decimal values range from 0 through 31, and from 128 through 159 are called control codes. Control codes often directly trigger particular actions. In Esker's emulator, control codes are configured in files with a .cod extension.

Here, for example, are the contents of the file ansi.cod:

```
nul 0
soh 0
stx 0
etx 0
eot 0
enq 0
ack 261
bel 250
bs 96
ht 99
lf 113
vt 0
ff 51
cr 97
so 0
si 0
dle 0
dc1 0
dc2 0
dc3 0
dc4 0
nak 0
syn 0
etb 0
can 0
em 0
sub 0
esc 0
fs 0
```

```
gs 0
rs 0
us 0
```

The first column contains the control code mnemonic and the second column contains the number of the action to be carried out on reception of the corresponding code. Three options can be used in the second column:

- Leave the value blank (the character is displayed on the terminal).
- Respond with 0 (no action is carried out, and the character isn't listed).
- Select an action from the list in the Escript.hlp file.

A control code can only execute a simple action.

> **Opening an existing .cod file**

Out of an emulation session

A .cod file is a text file that can be read in a Notepad-type text editor. You can therefore open a .cod file from this type of editor. The control code files supplied by Esker are copied into the emulator's installation directory.

During an emulation session

Select the Terminal option from the Settings menu.

Click on the .cod file chosen from the Control codes drop-down menu then click on the Modify button. The .cod file opens in the default text editor.

Code conversion

In some emulations, ASCII characters need to be converted before they can be sent correctly to the server. This conversion is managed by the .snd sent code files.

Here, for example, is an extract from file vt220.snd:

```
- \0xa1
> \0xa2
œ \0xa3
_ \0xa5
! \0xaa
® \0xab
ø \0xb0
ñ \0xb1
ý \0xb2
þ \0xb3
æ \0xb5
```

ã \0xb6

...

The left-hand column contains ASCII characters and the right-hand column contains the codes to be sent to the host machine.

> Opening an existing .snd file

Out of an emulation session

A .snd file is a text file that can be read in a Notepad-type text editor. You can therefore open a .snd file from this type of editor. The sent codes files provided by Esker are copied into the emulator's installation directory.

During an emulation session

Select the Terminal option from the Settings menu.

Click on the .snd file chosen from the Sent codes drop-down menu and then click on the Modify button. The .snd file opens in the default text editor.

Character tables

The character tables act as filters for displaying characters on the screen: An 8-bit character has 256 possible values. IBM-compatible micro-computers have their own encoding of these 256 characters. Certain characters are standard: 65 is represented as 'A', 66 as 'B', 48 as 'O', etc. Other characters, such as control characters, have a particular meaning for IBM. Different terminals have different character sets.

The objective is to define the tables for character representation. A representation is always encoded with 7 bits, that is, with the values 0 through 127.

The tables `ascii.tab` and `asciie.tab` (ASCII and the extended ASCII character set) represent the micro-computer codes 0 through 127 (`ascii.tab`) and 128 through 255 (`asciie.tab`). Other tables are:

| | |
|------------------|----------------------|
| UK.TAB | Britain |
| DECSU.TAB | DEC supplementary |
| DECSP.TAB | DEC special graphics |

A character table file looks like this:

```
80 81 82 83 84 85 86 87 88 89 8a 8b 8c 8d 8e 8f
90 91 92 93 94 95 96 97 98 99 9a 9b 9c 9d 9e 9f
a0 a1 a2 a3 a4 a5 a6 a7 a8 a9 aa ab ac ad ae af
b0 b1 b2 b3 b4 b5 b6 b7 b8 b9 ba bb bc bd be bf
c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 ca cb cc cd ce cf
d0 d1 d2 d3 d4 d5 d6 d7 d8 d9 da db dc dd de df
e0 e1 e2 e3 e4 e5 e6 e7 e8 e9 ea eb ec ed ee ef
```

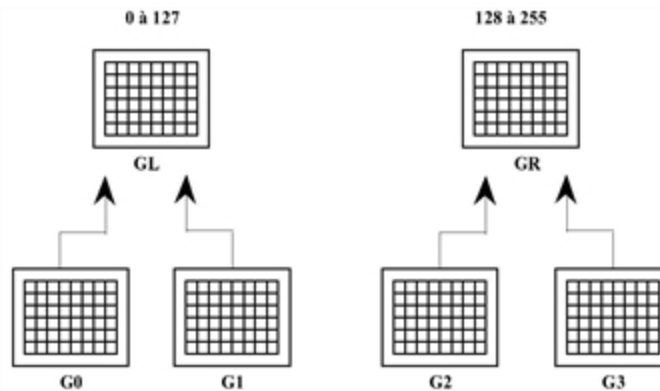
f0 f1 f2 f3 f4 f5 f6 f7 f8 f9 fa fb fc fd fe ff

There are 128 fields corresponding to the 128 possible arrangements of 7 bits. The horizontal axis shows the first four bits and the vertical axis shows the last three bits. This table only contains hexadecimal codes.

To change a particular character, its hexadecimal code must first be looked up in the ASCII table and then entered in the .tab file.

Internal character table management

To manage these tables during an emulation session, Esker's emulator uses a model based on VT100 emulation. There are several tables present in the memory of a VT100, but only 4 tables are available at any given time:



By default, one of the tables G0, G1, G2, or G3 is loaded into GL and GR. GL represents all the characters from 0 through 127, and GR, the characters from 128 through 255.

In Esker's emulator, there are four actions related to this procedure:

| Action | Description |
|--------|---------------------------------|
| 194 | Assigns a character table to G0 |
| 195 | Assigns a character table to G1 |
| 196 | Assigns a character table to G2 |
| 197 | Assigns a character table to G3 |

These actions are defined by a table number corresponding to those indicated in the terminal configuration box (Terminal option of the Session menu).

In Esker's emulator, eight actions allow you to fill GL and GR.:

| Action | Description |
|--------|---------------|
| 210 | lock G0 in GL |
| 211 | lock G1 in GL |

| Action | Description |
|--------|---------------|
| 212 | lock G2 in GL |
| 213 | lock G3 in GL |
| 214 | lock G0 in GR |
| 215 | lock G1 in GR |
| 216 | lock G2 in GR |
| 217 | lock G3 in GR |

Finally, four other simple actions let you access the next character in the tables G0, G1, G2, or G3 without using GL or GR:

| Action | Description |
|--------|---------------------|
| 218 | Selective use of G0 |
| 219 | Selective use of G1 |
| 220 | Selective use of G2 |
| 221 | Selective use of G3 |

This organization of 4 active tables (two of which are available by default) is complex. Most emulations have two permanent tables (GL and GR). The configuration file lets you use 10 alternative tables. You load one of these tables into GR or GL as follows:

Example:

194(4) 214

or with action labels:

TabAsG0(4) G0IntoGR

loads the 5th table into G0, then locks G0 into GR.

This organization lets you set the parameters to match virtually all existing terminal emulations.

Alternate character font

By default, PCs are only able to display 256 characters simultaneously. This limit poses some problems when trying to create emulations for more complex terminals that offer four or five different fonts.

In MS-DOS with a VGA or SVGA card, or in Windows, Esker's emulator can display 2 x 256 characters simultaneously by using an alternate font.

For a .tab file to be able to use an alternate font, simply replace the desired hexadecimal value by the number 1.

For example, the value 182 refers to the 130th position (82nd in hexadecimal) of the alternate font.

Using Esker actions

For its asynchronous emulator, Esker proposes a set of more than 350 actions, which can be combined within the text files for advanced configuration of terminal emulation. This chapter presents a few examples of how to use these actions.

The Escript.hlp file identifies the Esker actions by specifying their name, number and any parameters.

Note:

See the chapter "Advanced use of asynchronous emulator" in this manual for a detailed explanation of the configuration files described in the following examples.

Quitting Esker Viewer upon server request

Configured action 299 (labelled Exit) enables you to completely quit Esker Viewer hosting the ActiveX asynchronous emulation component

In ANSI emulation, this action is associated by default with the following escape sequence:

```
\033Q s 299(0)
```

The parameter associated with this action is the return code to be sent back by Esker Viewer to the program which started it. 299(1) is equivalent to exit(1) in a C program.

To enable the terminal emulator to interpret these escape sequences, simply add them (if they are not already there) to the escape sequences file corresponding to the type of terminal emulated.

Example:

The following shell script enables you to close Esker Viewer as if you were using the Quit option on the File menu:

```
echo "\033\033Q"
```

File transfer requested by server

Parametered actions Nos. 271 and 272 (labelled RcopyPut and RcopyGet) initiate file transfers between the PC and the server without user intervention.

In ANSI emulation, these actions are associated by default with the following escape sequences:

```
\033\033put%p0%s p 271
```

```
\033\033get%p0%s p 272
```

Parameter p0 associated with these actions is a character string designating the source file (s) and the destination file(s).

To enable the terminal emulator to interpret these escape sequences, simply add them (if they are not already there) to the escape sequences file corresponding to the type of terminal emulated.

You can also perform file transfer using macros (same RcopyPut and RcopyGet commands).

Windows to UNIX copy

The following escape sequence enables you transfer a PC Windows file to the UNIX server:

```
\033put%p0%s p 271
```

To enable the terminal emulator to interpret these escape sequences, simply add them (if they are not already there) to the escape sequences file corresponding to the type of terminal emulated.

Example:

The following shell script shows how to copy the file C:\autoexec.bat to the current directory of a UNIX session.

```
echo "Windows to Unix copy"
echo "\033\033put\"c:\\\\\\"autoexec.bat\"""
echo "Windows to UNIX copy finished "
```

Note:

The large number of '\' characters can be explained by the fact that the succession of command interpreters removes most of these despecializing characters.

UNIX to Windows copy

The following escape sequence enables you to transfer a file from the server to the PC running Windows:

```
\033get%p0%s p 272
```

To enable the terminal emulator to interpret these escape sequences, simply add them (if they are not already there) to the escape sequences file corresponding to the type of terminal emulated

Example :

The following shell script lets you transfer the file /etc/passwd to the root directory of the PC:

```
echo "UNIX to Windows copy"
echo "\033\033get\"/etc/passwd\"""
echo " UNIX to Windows copy finished"
```

PC programs started by server

Parametered action No. 296 (labelled ExecDOSProg) may be used to start Windows applications without user intervention. The program to execute is a parameter passed to these actions.

By default, this action is not associated with an emulation. To enable the terminal emulator to interpret the following escape sequence, simply add it to the escape sequences file corresponding to the type of terminal emulated:

```
\033X%p0%s p 296
```

Example:

The following shell script allows execution of the Write program:

```
echo "\033\033X\C:\\\\\\WINDOVS\\\\\\WRITE\""
```

Note:

The use of several \ characters is necessary because of the succession of command interpreters that strip most of these despecializing characters.

Macro execution requested by server

You can use parametered action No. 264 (labelled ExecMacro) to start a .mac type macro at the server's request, without user intervention.

In ANSI emulation, this action is associated by default with the following escape sequence:

```
\033M%p0%s p 264
```

The parameter p0 for this action is a character string indicating the command line you want to execute.

To enable the terminal emulator to interpret these escape sequences, simply add them (if they are not already there) to the escape sequences file corresponding to the type of terminal emulated.

Example:

The following shell script allows execution of macro sysadmsh.mac:

```
echo "\033\033M\"sysadmsh.mac\""
```

Transparent printing

In the character flow sent to the terminal emulator, the server can insert escape sequences to inform the emulator that the following characters are to be:

- Either displayed on the screen (conventional terminal emulation).
- Or sent directly to the printer connected to the PC emulating the terminal

This functionality lets the centralized application access the printer without using a buffer or print server.

Esker supplies a series of actions used to reroute the characters received from the server either to the PC screen, or to the printer, or to both simultaneously. To work in transparent print mode, the emulator must have escape sequences associated with these actions in the escape sequences configuration file.

Actions proposed by Esker

Action No. 260 (labelled SetDisplayOff) enables screen displays of the characters received to be inhibited.

Action No. 267 (labelled BeginPrint) enables the received characters to be re-routed to the printer using the Windows print manager.

Action No. 261 (labelled EndPrint) ends re-routing of the received characters to the printer.

Action No. 262 (labelled BeginRecvFile) enables a file to be received. Its parameter is the destination file name. This action can be used to send a file directly to an LPT or COM port: simply enter the name of the LPT or COM port as the action parameter.

Action No. 263 (labelled EndRecvFile) terminates reception of a file.

Action No. 259 (labelled SetDisplay) displays the characters received on the screen. This action is used when screen display of the characters has been inhibited (with the 260 Set-DisplayOff action).

Example 1: transparent printing on the PC's default printer via the Windows print manager

The following two escape sequences allow printing on the PC's default printer:

- [5i s 260 267
- [4i s 261 259

The first sequence involves switching to transparent print mode and starting printing on the PC's default printer of the characters sent by the server.

The second sequence involves terminating printing and inhibiting the transparent print mode so that the characters sent by the server can once again be displayed on the screen.

To enable the terminal emulator to interpret these escape sequences, simply add them (if they are not already there) to the escape sequences file corresponding to the type of terminal emulated. (for example ansi.seq for ANSI emulation).

Example:

The following is an example of using the UNIX shell to transparently print the file /etc/passwd using the default printer:

```
echo "Start transparent printing test (xxxx.seq)"
echo -n "\033[5i"
```

```
echo "File PASSWORD"
echo "-----"
cat /etc/passwd
echo "-----"
echo "End of file"
echo "\033[4i"
echo "End of test"
```

Example 2: Direct transparent printing on a printer port

The following two escape sequences allow direct printing on a printer port:

- [5i s 260 262 ("LPT1")
- [4i s 263 259

The first sequence involves switching to transparent print mode and starting printing by sending a file to port LPT1.

The second sequence involves ending transmission of the file and inhibiting transparent printing mode so that the characters sent by the server can once again be displayed on the screen.

To enable the terminal emulator to interpret these escape sequences, simply add them (if they are not already there) to the escape sequences file corresponding to the type of terminal emulated (for example ansi.seq for ANSI emulation).

Example:

In the following example, the UNIX shell transparently prints a copy of the file /etc/passwd on the LPT1 port of the PC:

```
echo "Start transparent printing test"
echo -n "\033[5i"
echo "Transparent copy test"
echo "-----"
cat /etc/passwd
echo "End of test"
echo "\033[4i"
```

Note:

If a Windows application prints at the same time as the emulator prints transparently, the print jobs will be mixed up because of the direct printing on the port.

Dynamically changing terminal type

Applications on UNIX hosts don't always use the same terminal type, even on the same server. Using one application after another in an emulation session can therefore present problems.

Action No. 270 (labelled ChangeTerminal) was designed to correct this situation. It allows the terminal type to be changed without closing the current session.

In ANSI emulation, this action is associated by default with the following escape sequence:

```
\033T%p0%s p 270
```

Parameter p0 associated with this action is a character string designating the type of terminal to be emulated (ANSI, VT320, IBM3151...).

To enable the terminal emulator to interpret these escape sequences, simply add them (if they are not already there) to the escape sequences file corresponding to the type of terminal emulated.

Example:

The following shell script enables the terminal to be changed to VT220:

```
echo "\033\033T\"VT220\""
```

Changing sessions automatically

Action No. 294 (labelled SetDisplaySession) allows the UNIX host to change the active terminal session without user intervention.

In ANSI emulation, this action is associated by default with the following escape sequence:

```
\033S%p0%1d p 294
```

Parameter p0 associated with this action is an integer between 0 and 31, designating the number of the session to be activated.

To enable the terminal emulator to interpret these escape sequences, simply add them (if they are not already there) to the escape sequences file corresponding to the type of terminal emulated.

Example:

The following shell script can be used to switch to session 2 and then session 1:

```
#To activate session No2
echo "\033\033S1"
#To activate session No1
echo "\033\033S0"
```

Mouse support in UNIX applications

Esker provides a number of actions for managing the mouse in traditional applications.

Principle

Esker's asynchronous emulator can send definable sequences each time a mouse event occurs, just as if a function key were pressed on the keyboard:

- Mouse movement
- Single or double click on right button.
- Release right button
- Single or double click on left button.
- Release left button
- Single or double click on middle button.
- Release middle button

The sequences that are sent always include the current mouse position in screen coordinates or virtual coordinates. To limit the data exchange across connections, UNIX applications can request that the terminal emulator send only certain events.

In addition, the application can control the mouse in the following ways:

- Display mouse
- Hide mouse
- Move mouse
- Return mouse status and position in a specified format
- Define the time interval in a “double-click”
- Define the time interval for sending mouse movement
- Return mouse status and position

Provided actions

The following actions are provided for mouse management:

- Initialize mouse.
- Deactivate mouse.
- Display mouse cursor.
- Hide mouse cursor.
- Move mouse cursor.
- Query mouse status.
- Activate mouse and select events.
- Define format of events expected by the application.

Mouse initialization

Action No. 277 (labelled InitMouse) is used to initialize the mouse.

In ANSI emulation, this action is associated by default with the following escape sequence:

```
\033Mi%p0%2d;%p1%2d;%p2%2dX p 277
```

To enable the terminal emulator to interpret these escape sequences, simply add them (if they are not already there) to the escape sequences file corresponding to the type of terminal emulated.

This action requires three parameters:

p0: (in integer format) is the event mask expected by the program:

- Move mouse 0x01
- Click left button 0x02
- Release left button 0x04
- Click right button 0x08
- Release right button 0x10
- Click middle button 0x20
- Release middle button 0x40
- Double click 0x80

p1: (integer format) is the time interval between mouse "reporting", expressed by the number of clock ticks (1 second = 18.2 ticks).

p2: (integer format) is the length of time taken by a double-click, expressed as the number of clock ticks (5 tends to be a good value).

Note:

If p2=0, the double-click is ignored.

Note:

Those actions associated with a double-click on a mouse button are run after any actions associated with a single click. This is standard Windows procedure.

Deactivate mouse

Action No. 278 (labelled ReleaseMouse) is used to deactivate the mouse.

In ANSI emulation, this action is associated by default with the following escape sequence:

```
\033Mc s 278
```

This action requires no parameters.

To enable the terminal emulator to interpret these escape sequences, simply add them (if they are not already there) to the escape sequences file corresponding to the type of terminal emulated.

Display mouse cursor

Action No. 279 (labelled ShowMouse) is used to display the mouse cursor in the application.

In ANSI emulation, this action is associated by default with the following escape sequence:

```
\033Md s 279
```

This action requires no parameters.

To enable the terminal emulator to interpret these escape sequences, simply add them (if they are not already there) to the escape sequences file corresponding to the type of terminal emulated.

Erase mouse cursor

Action No. 280 (labelled HideMouse) is used to hide the mouse cursor in the application.

In ANSI emulation, this action is associated by default with the following escape sequence:

```
\033Mh s 280
```

This action requires no parameters.

To enable the terminal emulator to interpret these escape sequences, simply add them (if they are not already there) to the escape sequences file corresponding to the type of terminal emulated.

Move the mouse cursor

Action No. 281 (labelled MoveMouse) is used to move the mouse cursor to a given position.

In ANSI emulation, this action is associated by default with the following escape sequence:

```
\033Mm%p0%d;%p1%dX p 281
```

To enable the terminal emulator to interpret these escape sequences, simply add them (if they are not already there) to the escape sequences file corresponding to the type of terminal emulated.

This action requires 2 parameters:

- p0: integer representing the new X position (columns).
- p1: integer representing the new Y position (rows).

Query mouse status

Action No. 282 (labelled QueryMouse) is used to obtain the status of the mouse (status of buttons plus position).

In ANSI emulation, this action is associated by default with the following escape sequence:

```
\033Mq s 282
```

This action requires no parameters.

To enable the terminal emulator to interpret these escape sequences, simply add them (if they are not already there) to the escape sequences file corresponding to the type of terminal emulated.

Activate mouse and select events

Action No. 283 (labelled ActivateMouse) is used to select certain mouse events.

In ANSI emulation, this action is associated by default with the following escape sequence:

```
\033Me%p0%dX p 283
```

p0: (integer format) is what the application expects to receive:

| | | |
|--------------|------|---|
| EVENT_ALL | 0x02 | Activates all the events defined by initialization |
| EVENT_MOVE | 0x01 | If (!EVENT_ALL) all events except mouse movement are returned if one of the buttons isn't pressed |
| XY_PHYSICAL | 0x04 | The current cursor position is returned in screen coordinates (virtual coordinates are the default) |
| XY_RELATIVE | 0x08 | Returns the position of the mouse cursor in relative coordinates to its previous position |
| LEFT_PANEL | 0x10 | Indicates that the left mouse button is reserved for the function-key panel |
| RIGHT_PANEL | 0x20 | Indicates that the right mouse button is reserved for the function-key panel |
| CENTER_PANEL | 0x40 | Indicates that the center mouse button is reserved for the function-key panel |

Format definition of events expected by the application

Action No. 284 (labelled DefineEventsSeq) is used to define the format of mouse events expected by the application.

In ANSI emulation, this action is associated by default with the following escape sequence:

```
\033Mf%p0%s p 284
```

This action follows the mouse initialization action. It takes one parameter:

p0: is a character string in C format that shows how to encode transmitted events. For example:

```
\033[Mf%d;%d;%d
```

The first parameter in this string is always the mouse status (see initialization constants). The next two parameters show the x and y positions of the cursor.

By default, the string format is:

```
%02x%03x%03x
```

Implementation

Mouse support is already configured for ANSI emulation with the following lines added to ansi.seq:

```
\033Mi%p0%2d;%p1%2d;%p2%2dX p 277
\033Mc s 278
\033Md s 279
\033Mh s 280
\033Mm%p0%d;%p1%dX p 281
\033Mq s 282
\033Me%p0%dX p 283
\033Mf%p0%s p 284
```

Using a mouse is too complicated to simulate with a simple UNIX command or shell script. For this reason, C source code (mouse.c) is provided by Esker in the installation directory. This file is all you need to implement a mouse interface for UNIX. Try compiling it and using it under emulation.

Miscellaneous Solutions

Color attributes in emulation

To make emulation more “attractive,” Tun EMUL lets you change the colors of classic screen attributes such as highlighting, underlining, etc. If you want to use different colors, you must change the Initialization line (the first line) of the appropriate “.seq” file.

Note:

A more simple solution consists of defining the display settings.

The colors you can use and their associated codes are:

```
0 : black
1 : blue
2 : green
3 : cyan
4 : red
5 : magenta
6 : brown
7 : light gray
8 : dark gray
9 : light blue
10 : light green
11 : light cyan
12 : light red
13 : light magenta
14 : light brown (yellow)
15 : white
```

Six different actions are used to control color selection:

| | |
|-----------|--|
| action 30 | color for normal video |
| action 31 | color for reverse video |
| action 66 | blink color |
| action 67 | underline color (since VGA displays are unable to provide underlining) |

| | |
|-----------|-----------------|
| | in text mode) |
| action 68 | highlight color |
| action 69 | dim color |

Action 30 is used with two parameters: The first determines character color; the second determines background color. For example, you can display white characters on a blue background with the parameters (15,1).

For color changes to take effect, enter the action number followed by the parameters between parentheses on the Initialization line of the .SEQ file, for example:

```
30 (7,1)
```

The same logic applies to action 31 (selection of reverse video color).

Action 66 is defined with only one hexadecimal parameter. If you want a white blinking character with a blue background, you can include the sequence: 66(0x71) on the Initialization line. The same applies to actions 67, 68, and 69.

These parameters (66, 67,68,69) are coded in one byte and are defined in hexadecimal notation. For example, if you want a light green blinking character on a light magenta background, use the corresponding hexadecimal codes:

```
0 : black
1 : blue
2 : green
3 : cyan
4 : red
5 : magenta
6 : brown
7 : light gray
8 : dark gray
9 : light blue
A : light green
B : light cyan
C : light red
D : light magenta
E : light brown (yellow)
F : white
```

Sample sequence using this notation: 66(0xAD).

132-column emulation under Windows

It's very easy to configure emulation with 132 columns in Windows. However, to see the entire screen, you must assign the Sys132PC font to the display settings file (".ctx").

Setting number of columns in a ".ctx" file

You can use display settings files to set the number of columns in an emulation session. Choose Settings > Display Settings > Terminal Font from the main menu.

Assigning 132 columns in the ".seq" file

For an emulation to always run with 132 columns, you can change the current ".seq" file. Modify the value of the parameter passed to action 1 as follows:

1(4): 132 columns on a monochrome VGA screen

1(5): 132 columns on a VGA color screen.

Action "1" determines the video display at startup. Usually the parameter is:

1(3): 80 columns on a VGA color screen.

Emulation with 25 lines

Most emulations are defined by default to display 24 lines on the screen. You can include the following action in the initialization string in the .seq file to specify the number of lines used:

```
5(0,23)
```

Action 5 defines the screen margins: The first parameter (0) designates the upper margin, and the second parameter (23) designates the lower margin. This action can be used in the initialization string as well as elsewhere in the .seq file.

To obtain an emulation with 25 lines, add 1 to the second parameter every time action 5 is defined. If it isn't present in the .seq file, insert it only in the initialization string. The following parameters cause emulation to use 25 lines:

```
5(0,24)
```

For example, the file wyse60.seq contains the following lines:

```
5(0,23) 62 72 (First line, initialization string)
\033
...
...
e( s 5(0,23)
e) s 5(0,24)
...
...
```

Change these lines to:


```
5(0,24) 62 72 (First line, initialization string)
\033
...
...
e( s 5(0,24)
e) s 5(0,25)
...
...
```

Scancode emulation

Some word processing programs in UNIX (Word, WordPerfect) need to use more keys than are usually provided by ordinary terminals. These programs also need the <Alt> keys to send values.

To handle this problem, these programs recommend the use of scancode emulations in which all the keys on a keyboard simply send their scancode, and not several different values.

Tun EMUL supports scancode emulation with actions 152 and 153. In standard ANSI emulation, these actions are associated with the following escape sequences:

```
\033~5
\033~4
```

Using scancode mode

Follow these steps to use the emulator in scancode mode:

1. Switch the emulator to scancode mode by sending the sequence `\033~5`
2. Change the tty on the UNIX host using this command:

```
stty isscancode xscancode
```

To switch back to Tun EMUL's native mode:

1. Send the sequence `\033~4`
2. Change the tty on the UNIX machine back to “normal” with the command:

```
stty -isscancode -xscancode
```

Using COM3 and COM4

Only COM1 and COM2 are completely standard on PCs. You can add two additional COM ports (COM3 and COM4) with proper definition of the IRQ and I/O addresses.

In Windows, COM3 and COM4 are defined using the Control Panel. Usually, ports COM3 and COM4 use the same IRQ as COM1 and COM 2, but have different I/O addresses (COM3=3E8 and COM4=2E8).

Defining modem commands

Tun EMUL uses the most standard Hayes commands for dialing modems. If these commands aren't sufficient for the proper use of your modem, choose Settings > Modem Commands... to define the codes that are required.

Program syntax

| | |
|----------------|---|
| EMUL32 | Asynchronous emulation in Windows |
| 3270_32 | Synchronous 3270 emulation module |
| 3287_32 | Synchronous 3287 printer emulation module |
| 5250_32 | Synchronous 5250 emulation module |
| 3812_32 | Synchronous 3812 printer emulation module |
| PANED32 | Function-key panel editor |

EMULWIN

EMUL32 is the command that starts the emulator.

Syntax

```
emul32 [config.cfg | config.cfz]
[-u [h][c][m][d][t][r][s][w]]
[-m in_macro] [-q out_macro]
[-h hostname] [-t term.ter]
[arg]
```

| | |
|------------|---|
| config.cfg | Configuration file (“.cfg” file). The suffix is optional. If used, this argument starts Tun EMUL using the supplied terminal session. |
| config.cfz | |
| -k | NIS resource (must be associated with -y) |
| -y | NIS table (host or session table, must be associated with -k). |
| -m | Followed by the name of a macro to be executed when emulation is started. |
| -q | Followed by the name of a macro to be executed when quitting emulation. |
| -h | Followed by the name of the host. Used with a config.cfg if config.cfg has @ as the host name. |
| -t | Terminal file (“.ter” file). Used with a config.cfg if config.cfg has @ as the terminal file. |
| -u | Used to restrict users' rights to emulation in Windows. It disables the fol- |

lowing options to ensure user security:

File > New Session...
File > Open...
File > Save
File > Save As...
File > Close
File > Print
File > Print Setup...
Settings, Connection, Transfers, Macro, Windows

The characters h, c, m, d, t, r, s, and w are used with -u to selectively reactivate features:

h reactivates File > Print Screen
c reactivates File > Print Setup... (with H).
m reactivates Macro
d reactivates Connection
t reactivates Transfers > Protocol Setup...
r reactivates Transfers > Get File...
s reactivates Transfers > Send File...
w reactivates Windows

arg

Represents the different parameters that are sent to the host machine when the emulation is loaded.

Example

```
EMUL32 conf1 -u
```

3270_32

3270 emulation in Windows

Syntax

```
3270_32 [-c config_file] [-h hostname] [-t term]  
[-k Niskey [-y Nistable]]
```

Description

3270_32.EXE is a Windows terminal emulator that provides 3270 emulation for connecting to IBM mainframes.

-c Starts emulation with an initialization file that defines:

| | |
|----|--|
| | Host name |
| | 3270 terminal type |
| | Color assignment |
| | Font size and name |
| | ASCII / EBCDIC correspondence |
| | Keyboard definition |
| -h | Followed by the name of the host |
| -t | Followed by the name of a terminal file |
| -k | NIS resource (must be associated with -y) |
| -y | NIS table (host or session table, must be associated with -k). |

You create initialization files by starting an emulation session, and then saving the current configuration with File > Save.

3287_32

3287 printer emulation in Windows

Syntax

```
3287_32 [-c config_file] [-h hostname] [-t term]
[-k Niskey [-y Nistable]]
```

Description

3287_32.EXE is a Windows printer emulator that enables the printing of IBM MainFrame terminal data to a printer connected to a Windows based PC.

| | |
|----|--|
| -c | Starts emulation with an initialization file that defines: |
| | Host name |
| | 3287 terminal type |
| | Color assignment |
| | Font size and name |
| | ASCII / EBCDIC correspondence |
| | Keyboard definition |
| -h | Followed by the name of the host |
| -t | Followed by the name of a terminal file |
| -k | NIS resource (must be associated with -y) |
| -y | NIS table (host or session table, must be associated with -k). |

You create initialization files by starting an emulation session, and then saving the current configuration with File > Save.

5250_32

5250 emulation in Windows

Syntax

```
5250_32 [-c config_file] [-h hostname] [-t term]
[-k Niskey [-y Nistable]]
```

Description

5250_32.EXE is a Windows terminal emulator that provides 5250 emulation for connecting to IBM AS/400 servers.

| | |
|----|---|
| -c | Starts emulation with an initialization file that defines: Host name 5250 terminal type Color assignment Font size and name ASCII / EBCDIC correspondence Keyboard definition |
| -h | Followed by the name of the host |
| -t | Followed by the name of a terminal file |
| -k | NIS resource (must be associated with -y) |
| -y | NIS table (host or session table, must be associated with -k). |

You create initialization files by starting an emulation session, and then saving the current configuration with File > Save.

3812_32

3812 printer emulation in Windows

Syntax

```
3812_32 [-c config_file] [-h hostname] [-t term]
[-k Niskey [-y Nistable]]
```

Description

3812_32.EXE is a Windows printer emulator that enables the printing of IBM AS/400 terminal data to a printer connected to a Windows based PC.

| | |
|----|---|
| -c | Starts emulation with an initialization file that defines: Host name 3812 terminal type Color assignment Font size and name ASCII / EBCDIC correspondence Keyboard definition |
| -h | Followed by the name of the host |
| -t | Followed by the name of a terminal file |
| -k | NIS resource (must be associated with -y) |
| -y | NIS table (host or session table, must be associated with -k). |

You create initialization files by starting an emulation session, and then saving the current configuration with File > Save.

PANED32

Function-key panel editor.

Syntax

```
PANED32 [file]
```

Description

PANEDIT is a Windows program that creates and customizes key panels for use with Tun EMUL emulators. The only argument you can pass to the program is the name of a key panel file, extension “.pan”, that contains the key panel definition.

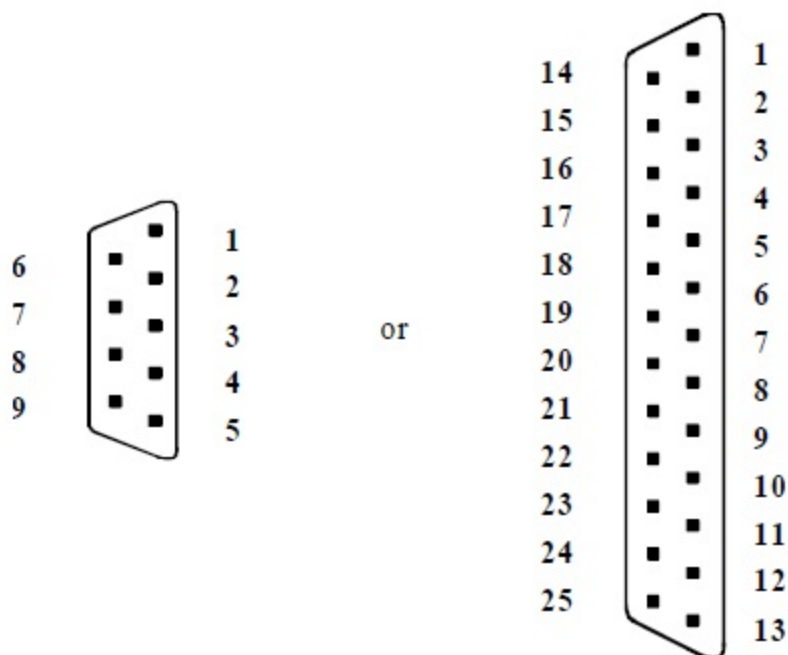
A

Serial Cabling

Many problems with serial connections are due to cabling errors. This section will help you construct the proper serial cable for your situation.

Overview of serial ports and cables

Serial ports on both PCs and servers typically consist of a 9- or 25-pin male connector, with the pins numbered as shown below:



Depending on the pin assignment of the serial (RS232) connector, computers are classified into two categories:

- Terminals.
- Host machines.

DTE/DCE

Serial ports can be configured as either DTE (Data Terminal Equipment) or DCE (Data Communication Equipment). The standard adopted by IBM identifies a port by the type of connector:

- male: DTE configuration
- female: DCE configuration

Typically, PCs are configured as DTE, but they can be used either as terminals (running emulation software) or as host machines (running a multi-user operating system).

Pin assignment varies depending on whether the serial port is configured as DTE or DCE:

| Signal | DTE 25 pins | DCE 25 pins |
|------------------------------|-------------|-------------|
| Transmit Data (TD) | 2 | 3 |
| Receive Data (RD) | 3 | 2 |
| Request to send (RTS) | 4 | 5 |
| Clear to send (CTS) | 5 | 4 |
| Data station ready (DSR) | 6 | 20 |
| Ground (GRD) | 7 | 7 |
| Data Carrier Detection (DCD) | 8 | 8 |
| Data terminal ready (DTR) | 20 | 6 |
| Recall indicator (RI) | 22 | 22 |

Minimum Cabling

At least three wires must be used in a direct connection between a PC running **Tun EMUL** and a host machine:

| PC 25-pin male DTE | | Host Machine 25-pin female |
|--------------------|---|----------------------------|
| TD 2 | ↔ | 2RD |
| RD 3 | ↔ | 2 TD |
| GRD 7 | ↔ | 7GRD |

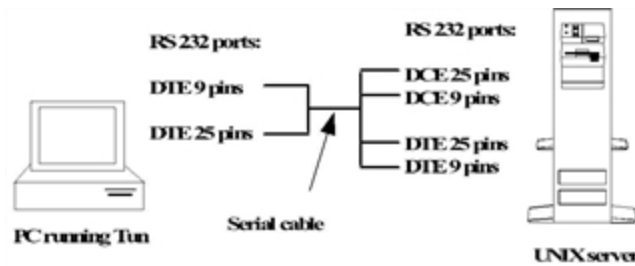
9-to-25 pin conversion

To correctly convert a 9-pin connector to a 25-pin connector, follow the diagram below:

| 9-pin female connector | | 25-pin male connector |
|------------------------|---|-----------------------|
| 1 | ↔ | 8 |
| 2 | ↔ | 3 |
| 3 | ↔ | 2 |
| 4 | ↔ | 20 |
| 5 | ↔ | 7 |
| 6 | ↔ | 6 |

| 9-pin female connector | | 25-pin male connector |
|------------------------|---|-----------------------|
| 7 | ↔ | 4 |
| 8 | ↔ | 5 |
| 9 | ↔ | 22 |

The possible combinations of serial connections are summarized in the figure below:



Note:

In some cases, the serial ports on the host machine can be configured as DTE. This is the case for UNIX and XENIX operating systems that run on micro-computers.

Pins 2 and 3 need to be crossed when connecting the PC running **Tun EMUL** to this type of host machine (from DTE to DTE).

Cable possibilities

The different possible (minimum) cabling configurations are shown in the following diagrams.

| PC | | Host-Computer |
|------------|---|---------------|
| 25-Pin DTE | | 25-Pin DCE |
| 2 | ↔ | 2 |
| 3 | ↔ | 3 |
| 7 | ↔ | 7 |
| 25-Pin DTE | | 9-Pin DCE |
| 2 | ↔ | 3 |
| 3 | ↔ | 2 |
| 7 | ↔ | 5 |
| 25-Pin DTE | | 9-Pin DCE |
| | ↔ | 2 |
| 2 | ↔ | 3 |

| PC 25-Pin DTE | | Host-Computer 25-Pin DCE |
|------------------|---|-----------------------------|
| 7 | ↔ | 7 |
| 25-Pin DTE | | 25-Pin DTE |
| 2 | ↔ | 2 |
| 3 | ↔ | 3 |
| 7 | ↔ | 5 |
| 9-Pin DTE | | 25-Pin DCE |
| 3 | ↔ | 2 |
| 2 | ↔ | 3 |
| 5 | ↔ | 7 |
| 9-Pin DTE | | 9-Pin DCE |
| 3 | ↔ | 3 |
| 2 | ↔ | 2 |
| 5 | ↔ | 5 |
| 9-Pin DTE | | 25-Pin DTE |
| 3 | ↔ | 3 |
| 2 | ↔ | 2 |
| 5 | ↔ | 7 |
| 9-Pin DTE | | 9-Pin DTE |
| 3 | ↔ | 2 |
| 2 | ↔ | 3 |
| 5 | ↔ | 5 |

B

Macro Examples

The following macro automates connection, login, and automatic transfer of selected files (vt220.* and ansi.seq) from the to the /tmp directory of the UNIX host.

```
# Characters sent by Host computer not displayed
Title "File Transfer"

# Start
Label BEGIN
# Read password
ShowMessage
ReadPasswd "Enter your password : " PASSWD
HideMessage

# Make connection
Repeat 3
# Send carriage-return character
Repeat 5
SendAndReceive 5 "\n" "ogin"
IfNoError break
end
IfError NOCONNECTION

# Send login
SendAndReceive 9 "root\n" "assword" "# "
IfError continue
IfEqual "# " break
SendAndReceive 9 "%PASSWD\n" "# " "ogin:" "ERM"
IfError continue
IfEqual "# " break

# Return to start of program if login incorrect
IfEqual "ogin:" BEGIN
# Set the TERM variable if necessary
SendAndReceive 9 "\n" "# "
IfError continue
IfEqual "# " break
end
# Start application
Set FILE "VT220.*"

RcopyPut c:\\TUN\\EMUL\\VT220.* /tmp
IfError TRANSFERERROR
Set FILE "ansi.seq"
RcopyPut c:\\TUN\\EMUL\\ansi.seq /tmp
IfError TRANSFERERROR
```

```

# Transfer done
ShowMessage
ClearMessage
Echo -c "Transfer Done"
Sleep 3

# Exit from the emulator
Exit

#Transfer error
Label TRANSFERERROR
ShowMessage
Echo "Error Transferring %FILE"
ReadVar "Press Return to quit" ANSWER
# Exit the emulator
exit

# No login
Label NOCONNECTION
ShowMessage
Echo "Communication failed"
ReadVar "Press Return to quit" ANSWER
# Exit the emulator
exit

```

This next macro automates connection, login, and then starts the system maintenance utility “sysadmsh” on a SCO UNIX host:

```

# Characters sent by Host computer not displayed
Hide

# Start
Label BEGIN
# Read login and password
ReadVar "Enter your user name : " USER
ReadPasswd "Enter your password : " PASSWD

# Make connection
Repeat 3
# Send carriage-return character
Repeat 5
SendAndReceive 5 "\n" "ogin"
IfNoError break
end
IfError NOCONNECTION

# Send login
SendAndReceive 9 "%USER\n" "assword" "# "
IfError continue
IfEqual "# " break
SendAndReceive 9 "%PASSWD\n" "# " "ogin:" "ERM"
IfError continue
IfEqual "# " break

# Return to start of program if login incorrect

```

```

    IfEqual "ogin:" BEGIN
# Set the TERM variable if necessary
SendAndReceive 9 "\n" "# "
IfError continue
IfEqual "# " break
end
# Start application
#Send "sysadmsh\n"
Send "# Starting the program \033[35h\n"
# Display received characters
Display
# Return to the emulator
Return
# No login
Label NOCONNECTION
Echo "Communication failed"
ReadVar "Press Return to quit" ANSWER
# Exit the emulator
exit

```

Note:

You can substitute the name of another UNIX application or shell script for "sysadmsh."

The next macro is an example of remote communication via modem. It connects the user to Esker's BBS. It assumes a modem is connected to the serial port of a host running SCO UNIX and that the following lines are included in the /usr/lib/uucp/Devices file on the UNIX machine.

```

ACU tty1a - 1200 hayes1200 \D
ACU tty1a - 9600 hayes9600 \D

```

You can elaborate the basic macro by adding the loop required to test the presence of several modems or ports, and choose the next available one).

```

# Start
Label BEGIN

# Send login
Hide
SendAndReceive 20 "root\r" "assword"
IfError NOCONNECTION
SendAndReceive 20 "support\r" "TERM"

```

```
IfError NOCONNECTION
SendAndReceive 20 "\r" "#"
IfError NOCONNECTION
send "clear\r"
pause 2

# Check if the port is busy

ReadVar "Enter the port you want to use: " PORT

ClearMessage
Echo "Please wait, ... connecting ... , this will take a few seconds"
SendAndReceive 3 "cu -s9600 -l %PORT dir\r" "connected" "LOCKED"

# You can insert the name of the device directly, if known
# SendAndReceive 20 "cu -l /dev/ttyla dir\r" "connected" "LOCKED"
# Then you do not need the variable PORT

IfEqual "LOCKED" BUSY

# Connection
Send "atz\r"
pause 4
Send "atdt 0,72449472\r"
sleep 20
HideMessage

Display

Receive 0 "Thank you" "NO CARRIER"
pause 1
exit

# No connection
Label NOCONNECTION
Echo "Communication failed"
ReadVar "Press Return to quit: " ANSWER
```

```
# Exit the emulator
```

```
exit
```

```
Label Busy
```

```
ClearMessage
```

```
ShowMessage
```

```
echo "MINITEL IS ALREADY IN USE, PLEASE TRY AGAIN LATER"
```

```
sleep 5
```

```
ReadVar "Press return to quit: " ANSWER
```

```
#exit the emulator
```

```
exit
```


Index

.

.COD 149

.TAB 151

1

132 Column emulation
Windows 168

3

3270
Multiple connection 52
TN3270 48-49
TN3270E 48-49

3270.EXE 172

3287
Emulation 102
Options 108

3287 printing
Associated printer 105
Close session on confirmation 104
Host name 104
Port 105
Printer logical unit 105
Reconnection after a
shutdown 104
Startup Connection 104
Terminal logical unit 105
Terminal type 104

3287.EXE 173

3812
Emulation 102
Options 108

3812 printing
Close session on confirmation 104
Device name 106-107
Host name 106-107
Library name 106, 108
Local LU name 107
Mode name 107
Password 107
Port 106
Queue name 106, 108
Reconnection after a
shutdown 104
Startup Connection 104
Terminal type 105, 107
User name 107

3812.EXE 174

5

5250
Multiple connection 52
System name 51
Terminal type 51
TN5250 Connection 49
Unit name 51

A

Actions
Dynamically changing terminal
type 160
File transfer 155
Quitting emulator 155

Additional resources (asynchronous
emulation) 26

Alignment (asynchronous emulation)
65

Alternate character font 153

API
e32appc.dll 43
wappc32.dll 44

API APPC
e32appc.dll 44

APL 57
Characters 57
Keyboard 57
Mode 57

AS/400 39

ASCII (File transfer protocol) 79

ASCII/EBCDIC Conversion (file
transfer in synchronous emulation) 81

Associated printer (3287) 105

Asynchronous emulation
.cfg files 27
.cfz files 25
.cod files 138
.ctx files 27
.fun files 138
.key files 137
.nat files 138
.seq files 138
.snd files 138
.tab files 138
.ter files 137
Additional resources 26
Alignment 65
Archive 26
ASCII (file transfer protocol) 79
Attribute style 70
Baud Rate 19
Bitmap file 65
Caps lock 94
Capture the attributes 68
Center terminal 65
Character attributes 67
Character tables 138
Code conversion 138

Combination of attributes 68
Configurations 25
Control codes 138
Copy options 78
CR/LF to LF conversion 80
Cursor coupling 64
Customization 29
Data bits 19
DDE 30
Definition 12
Display zone 65
Docking the function-key panel 71
Dynamic sizing 66-67
Encryption in macros 96
Escape sequences 138
Esker emulations 13
Firewall 100
Fit to display zone 65
Font 66-67
Foreground/Background colors 68
Function-keys 138
Function key labels 89, 122
Keyboard 137
Keyboard selection 86
Lock state 87, 94
Macro encryption library 97
Mapping a function-key to a key-
board key 89
Mapping an instruction to a mouse
event 72
MDI mode 25
Menu customization 16
Menus 16
Mnemonics 88, 90, 122
Mouse configuration 72
National files 87
National keyboard 29, 138
Num lock 94
Options 28
Other functions using the
mouse 73
Parity 20
Paste and send 78
Resources editor 26
Revamping 27
Rtunplus (File transfer protocol)
79
Running a macro using the key-
board 89
Running a macro using the
mouse 73
Running a script from the
keyboard 73, 89
Scan code 87
Screen 64
Screen size 64
Scroll bars 64
Scroll type 64
Scroll with text 65
Selecting a mouse event 72
Sending a character string using
the keyboard 88
Sending a function-key using the
mouse 73
Sending a string using the
mouse 72
Serial Port 19
Session 25

- Session on TCP/IP network 18
- Session over serial lines 19
- Settings files 15
- Show keyboard 94
- Standard resources 26
- Status bar 18
- Stop bits 20
- Terminal 14, 137
- Terminal display 27
- Tile 65
- Toolbar 17
- Windows/Iso 8859 conversion 80
- X-Modem (File transfer protocol) 79
- Y-Modem (File transfer protocol) 79
- Z-Modem (File transfer protocol) 79

- Asynchronous Emulation
 - Context 14
 - Revamping 14

- Attribute capture (synchronous emulation) 69

- Attribute filters (hotspots) 115

- Attribute style (asynchronous emulation) 70

- Auto-active (hotspots) 112-113
 - Events 120

B

- Background colors (asynchronous emulation) 68

- Background colors (synchronous emulation) 69

- BackOffice 42, 44

- Baud rate 19

- Baud Rate (emulation on serial lines) 19

- Bitmap file (asynchronous emulation) 65

- BULL 15

- Button (hotspots) 112
 - Events 119

C

- Caps lock (asynchronous emulation) 87, 94

- Caps Lock key (synchronous emulation) 84

- Capture the attributes (asynchronous emulation) 68

- Character attributes (asynchronous emulation) 67

- Character tables (asynchronous emulation) 138

- Character tables (synchronous emulation) 86

- Characters
 - Character mode 12
 - Data stream 12

- Clear selection (synchronous emulation) 78

- Close session on confirmation 48, 104

- Code conversion (asynchronous emulation) 138

- Color selection 166

- Colors
 - Background colors (synchronous emulation) 69
 - Customization 67
 - Foreground colors (synchronous emulation) 69

- Combination of attributes (asynchronous emulation) 68

- Combined list (hotspots) 113
 - Options 120

- Command
 - CMS/TSO (file transfer in synchronous emulation) 81

- COMx Ports 19

- Configuration files
 - .seq 138, 168

- Configuration in asynchronous emulation 25

- Connection (3287 emulation) 104

- Connection (3812 Emulation) 104

- Connection (synchronous emulation) 47

- Connection 3270
 - Close session on confirmation 48
 - Host name 48
 - Logical unit name 49, 105
 - Multiple 52
 - Netware for SAA LUA 49
 - Port 48
 - Reconnection after a shutdown 48
 - SNA Server FMI 49
 - SNA Server LUA 49
 - Startup Connection 48
 - Terminal type 48-49
 - TN3270 48
 - TN3270E 48

- Connection 3287
 - Netware for SAA LUA 105
 - SNA Server FMI 105
 - SNA Server LUA 105
 - Terminal type 105
 - TN3270E 104

- Connection 3812
 - Netware for SAA 107
 - Password 107
 - SNA Server 107
 - TN5250E 105
 - User name 107

- Connection 5250
 - Close session on confirmation 48
 - Host name 50
 - Local LU Name 51
 - Mode name 51
 - Multiple 52
 - Netware for SAA 51
 - Password 52
 - Port 50
 - Reconnection after a shutdown 48
 - SNA Server 51
 - Startup Connection 48
 - Terminal type 49
 - TN5250 49
 - User name 52

- Context (asynchronous emulation) 14

- Control codes (asynchronous emulation) 138

- Controls (hotspots) 112
 - Auto-active 113
 - Button 112
 - Combined list 113
 - Menu 112
 - Scrollbar 113
 - Text 113
 - Transparent text 113

- Conversion (file transfer in synchronous emulation) 82

- Conversion Windows/Iso 8859 (file transfer in asynchronous emulation) 80

- Copy options (asynchronous emulation) 78

- Copy/Paste 77

- CR/LF Conversion (file transfer in asynchronous emulation) 80

- Ctrl key (synchronous emulation) 84

- Cursor coupling (asynchronous emulation) 64

- Cursor Position 55

- Cursor style (synchronous emulation) 85

- Customization
 - Emulation customizer 98
 - Toolbar 17

- Customizing the asynchronous emulator 29

D

- Data bits (emulation on serial lines) 19

DDE

- Application 30
- Item 30
- Topic 30

- DDE (asynchronous emulation) 30

- Debug 140

- DEC 15

Display

- Full screen mode 17
- Menu display 17
- Toolbar display 17

- Display zone (asynchronous emulation) 65

Docking

- Dockable toolbars 17
- Toolbar docking 45

- Docking the function-key panel (asynchronous emulation) 71

- Docking the function-key panel (synchronous emulation) 71

- DTE/DCE 176

- Dynamic sizing (asynchronous emulation) 66-67

E

- EBCDIC 86

Emulation

- 132 columns 66

Emulation 3270

- Multiple connection 52
- Netware for SAA LUA connection 49
- SNA Server FMI connection 49
- SNA Server LUA connection 49
- TN3270 connection 48
- TN3270E connection 48

Emulation 3287

- Connection 104
- Logical unit name 105
- Netware for SAA LUA connection 105
- SNA Server FMI connection 105

- SNA Server LUA connection 105
- Terminal type 105
- TN3270E connection 104

Emulation 3812

- Connection 104
- Netware for SAA connection 107
- SNA Server connection 107
- TN5250E connection 105

Emulation 5250

- Multiple connection 52
- Netware for SAA connection 51
- SNA Server connection 51
- TN5250 Connection 49

Emulation screen

- Customization 64

- EMULWIN.EXE 171

- Encryption in macros (asynchronous emulation) 96

- Escape sequence headers 140

Escape sequences 138

- Data stream 12

- Escape sequences (asynchronous emulation) 138

Events

- Auto-active (hotspots) 120
- Button (hotspots) 119
- Hotspots 121
- Scrollbar (hotspots) 120
- Text (hotspots) 119
- Transparent text (hotspots) 119

F

- File reception 139

- File transfer 78, 155

Files

- .bmp files 65
- .cfg files 16, 25, 27
- .cfz files 16, 25-26
- .cod files 15, 138
- .ctx file 14
- .ctx files 15, 27
- .fky files 90
- .fun files 15, 89, 138
- .hot files 16
- .kbd files 96
- .kbm files 87
- .kbt files 87
- .key files 87, 94, 137
- .mac files 89, 96
- .nat files 15, 87, 138
- .pan files 16
- .ptp files 76
- .seq files 15, 138
- .ses files 15
- .snd files 15, 138
- .tab files 15, 138
- .ter files 15, 137

- e32appc.dll 43-44
- File .fun 122
- File .mac 122
- Log file 111
- National files 87
- wappc32.dll 44
- Winrui32.dll 41-42

- Final text (printing with templates) 76

- Firewall (asynchronous emulation) 100

- Fit to display zone (asynchronous emulation) 65

- Flat style 45

- FMI 39

- Font (asynchronous emulation) 66-67

- Font selected when printing text 74

- Foreground colors (asynchronous emulation) 68

- Foreground colors (synchronous emulation) 69

- Frame (synchronous emulation) 66

- Function-key labels (asynchronous emulation) 89, 122

- Function-key panels 123
- Show or hide 70

- Function-keys (asynchronous emulation) 138

G

Gateways

- Microsoft SNA Server 42, 44
- Microsoft SNA Server gateway 39
- Netware for SAA 13
- Netware for SAA gateway 39
- Novell Netware for SAA 43
- SNA 13
- UNIX SNA-TCP/IP (3270) 41
- UNIX SNA-TCP/IP (5250) 43
- UNIX SNA-TCP/IP gateway 39

- Gateways using 3270 40

- Gateways using 5250 43

H

- Highlight 167

- HLLAPI 58

- Host Command (File transfer in synchronous emulation) 81

Host name
 3287 emulation 104
 3812 emulation 106

Host System (file transfer in synchronous emulation) 80

Hotspots 83, 112
 Controls 83, 112
 Running a macro 122
 Signature 112
 Type auto-active 112

I

IBM Printers emulation 101

Image (synchronous emulation) 66

Initial Action (file transfer in synchronous emulation) 81

Initial page (printing with template) 76

Initial text (printing with templates) 76

Interactive Unix 15

K

Keyboard
 Assigning actions (synchronous emulation) 95
 Associating a mnemonic (asynchronous emulation) 90, 122
 Customization 86
 Mapping a function-key (asynchronous emulation) 89
 Running a macro (asynchronous emulation) 89
 Running a script (asynchronous emulation) 73, 89
 Sending a character string (asynchronous emulation) 88

Keyboard (asynchronous emulation) 29, 137

L

Lock state (asynchronous emulation) 87, 94

Log 83

Log File 83, 111

Logical unit 49, 51, 101, 106-108

Long name in HLLAPI (synchronous emulation) 85

LU 40, 101

LU (3270) 49

LU (3812) 107

LU (5250) 51

LU6.2 43-44

LUA 39

M

Macro encryption library (asynchronous emulation) 97

MainFrame 39

MDI 25

Menu (hotspots) 112
 Options 119

Menus 16, 44
 Customization 16, 44, 98

Microsoft SNA Server 39, 42, 44

Minitel 15

Mnemonics (asynchronous emulation) 88, 90, 122

Mode (file transfer in synchronous emulation) 82

Modem 14

Modem commands 170

Mouse
 Configuration 72
 Mapping an instruction 72
 Other functions (asynchronous emulation) 73
 Running a macro (asynchronous emulation) 73
 Selecting an event (asynchronous emulation) 72
 Sending a function-key (asynchronous emulation) 73
 Sending a string (asynchronous emulation) 72

Moving the cursor (synchronous emulation) 85

Multiple (3270, 5250 connection) 52
 Random selection 52
 Timeout 52

Multiple connection (3270, 5250) 52
 Random selection 52
 Timeout 52

Multiple file transfer (synchronous emulation) 82

Multiple sessions 25

N

National keyboard (asynchronous emulation) 29, 138

Netware for SAA 39, 41
 APPC (5250) gateways 43
 Esker emulations 13

Netware for SAA (3812)
 Device name 106-107
 Host name 107
 Library name 106, 108
 Mode name 107
 Password 107
 Queue name 106, 108
 Terminal type 107
 User 107

Netware for SAA (5250) 51, 107
 Mode name 51
 Password 52
 User 52

Netware for SAA Gateways 3270 41
 Native mode (3270) 40

Netware for SAA Gateways (5250) 43

Netware for SAA LUA (3270) 49, 105
 Terminal type 49

Netware for SAA LUA (3287)
 Logical unit name 105
 Terminal type 105

Next page key (printing with template) 76

Nixdorf 15

Norms
 Winsock 14

Novell Netware for SAA 41, 43

Num lock (asynchronous emulation) 87, 94

Numeric fields (synchronous emulation) 85

O

OIA (Operator Information Area) 53
 APL mode 55
 bar 46
 Content 53
 Insert mode 55
 Shift mode 55

Operator Information Area 53

Options
 Combined list (hotspots) 120
 Hotspots 121
 Menu (hotspots) 119

Options (3287 printing) 108

Options (3812 printing) 108

P

Packet Mode 39

Packet Size (File transfer in synchronous emulation) 81

PANEDIT.EXE 175

Parity (emulation on serial lines) 20

Paste and send (asynchronous emulation) 78

PC keyboard locking 54

Philips 15

Port 19

3287 emulation 105

3812 emulation 106

Previous page key (printing with template) 76

Print area (printing with templates) 76

Print options (3287) 108

Print options (3812) 108

Printer emulation

Esker emulations 13

Printer logical unit (3287) 105

Printers

3287 13

3812 13

Printing

Final text 76

Initial page 76

Initial text 76

Previous page/Next page 76

Print area 76

Print screen 74

Printing with templates 75, 77

Printing (3287)

Close session on confirmation 104

Reconnection after a

shutdown 104

Startup Connection 104

Printing (3812)

Close session on confirmation 104

Reconnection after a

shutdown 104

Startup Connection 104

Profile (emulation customization) 98

Protocols

ASCII 79

LU6.2 protocol 43-44

Rtunplus 79

Telnet 13

Telnet 3270 13

Telnet 5250 13

Telnet protocol 14, 18

TN3270 protocol 39

TN3270E protocol 39

TN5250 protocol 39

X-Modem 79

Y-Modem 79

Z-Modem 79

Proxy 100

Q

Queue Name (3812) 106, 108

R

Random selection 52

Recognition of a string (hotspots) 83

Reconnection after a shutdown 48, 104

Regular expression (hotspots) 115

Resources editor (asynchronous emulation) 26

Revamping (asynchronous emulation) 14, 27

RS232 14, 19

Rtunplus (File transfer protocol) 79

Rule cursor (synchronous emulation) 85

Running a macro on a hotspot 122

S

Scan code (asynchronous emulation) 87

SCO UNIX 15

SCO XENIX 15

Screen

Screen mode 12

Screen (asynchronous emulation) 64

Screen Mode 39

Screen size (asynchronous emulation) 64

Script edition

Running a script from the keyboard 73, 89

Scripting

Hotspots 120

Scroll bars (asynchronous emulation) 64

Scroll type (asynchronous emulation) 64

Scroll with text (asynchronous emulation) 65

Scrollbar (hotspots) 113

Events 120

Search area (hotspots) 115

Search attributes (hotspots) 115

Serial cable possibilities 178

Serial cabling 176

Serial lines 19

Serial link

RS232 14

Serial Port (emulation on serial lines) 19

Server system (file transfer in synchronous emulation) 82

Servers

IBM AS/400 Server 39

IBM MainFrame Server 39

Session 18-19, 25

Session 3270

Close session on confirmation 48

Host name 48

Logical unit name 49

Multiple connection 52

Netware for SAA LUA connection 49

Port 48

Reconnection after a shutdown 48

SNA Server FMI connection 49

SNA Server LUA connection 49

Startup Connection 48

Terminal type 48-49

TN3270 connection 48

TN3270E connection 48

Session 3287

Associated printer 105

Close session on confirmation 104

Host name 104

Netware for SAA LUA connection 105

Port 105

Printer logical unit 105

Reconnection after a

shutdown 104

SNA Server FMI connection 105

SNA Server LUA connection 105

Startup Connection 104

Terminal logical unit 105

Terminal type 104-105

- TN3270E connection 104
- Session 3812
 - Close session on confirmation 104
 - Device name 106-107
 - Host name 106-107
 - Library name 106, 108
 - Local LU name 107
 - Mode name 107
 - Netware for SAA connection 107
 - Password 107
 - Port 106
 - Queue name 106, 108
 - Reconnection after a shutdown 104
 - SNA Server connection 107
 - Startup Connection 104
 - Terminal type 105, 107
 - TN5250E connection 105
 - User name 107
- Session 5250
 - Close session on confirmation 48
 - Host name 50
 - Local LU Name 51
 - Mode name 51
 - Multiple connection 52
 - Netware for SAA connection 51
 - Password 52
 - Port 50
 - Reconnection after a shutdown 48
 - SNA Server connection 51
 - Startup Connection 48
 - Terminal type 49
 - TN5250 Connection 49
 - User name 52
- Short name in HLLAPI (synchronous emulation) 85
- Siemens 15
- Signature 112
- SNA 39
 - Esker emulations 13
- SNA Gateways
 - Novell Netware for SAA 41
 - UNIX SNA-TCP/IP 41, 43
- SNA Server 44
 - APPC (5250) gateways 43
 - Gateways 42
- SNA Server (3812)
 - Device name 106-107
 - Host name 107
 - Library name 106, 108
 - Local LU name 107
 - Mode name 107
 - Password 107
 - Queue name 106, 108
 - Terminal type 107
 - User 107
- SNA Server (5250) 51, 107
 - Local LU name 51
 - Mode name 51
 - Password 52
- User 52
- SNA Server Client 44
- SNA Server FMI (3270) 49, 105
 - Terminal type 49
- SNA Server FMI (3287)
 - Logical unit name 105
 - Terminal type 105
- SNA Server gateways
 - Native mode (3270) 40
- SNA Server LUA (3270) 49, 105
 - Terminal type 49
- SNA Server LUA (3287)
 - Logical unit name 105
 - Terminal type 105
- Standard resources (asynchronous emulation) 26
- Startup Connection 48, 104
- Status bar 18, 46
- Stop bits (emulation on serial lines) 20
- String recognition (hotspots) 112
- Synchronous emulation
 - .cfg files 25
 - .cfz files 26
 - ASCII/EBCDIC Conversion 81
 - Assigning actions 95
 - Attribute capture 69
 - Caps Lock key 84
 - Character tables 86
 - Clear selection 78
 - CMS/TSO command 81
 - Connection 47
 - Ctrl key 84
 - Cursor 85
 - Cursor style 85
 - Definition 12
 - Docking the function-key panel 71
 - Esker emulations 13
 - Foreground/Background colors 69
 - Frame 66
 - Host Command 81
 - Host System 80
 - Image 66
 - Initial action 81-82
 - Long name in HLLAPI 85
 - Menu customization 44
 - Menus 44
 - Mode 82
 - Moving the cursor 85
 - Multiple file transfer 82
 - Numeric fields 85
 - OIA bar 46
 - Packet Size 81
 - Printers emulation 101
 - Rule cursor 85
 - Server system 82
 - Short name in HLLAPI 85
 - Status bar 46
- Timeout 80
- Toolbar 45
- Synchronous Emulation
 - Working principles 39
- Sys132PC 66

T

- TCP/IP 18
- Telnet 13, 18
- Telnet 3270 13
- Telnet 5250 13
- Template
 - Printing with templates 75
- Terminal
 - Customization 84
 - Definition 12
 - Initialization 140
- Terminal (asynchronous emulation) 137
- Terminal centering (asynchronous emulation) 65
- Terminal definition (asynchronous emulation) 14
- Terminal display (asynchronous emulation) 27
- Terminal emulation
 - 3270 emulation 39
- Terminal initialization 167
- Terminal logical unit (3287) 105
- Terminal type 160
 - 3287 emulation 104
 - 3812 emulation 105
- Text (hotspots) 113
 - Events 119
- Tile (asynchronous emulation) 65
- Timeout (3270, 5250 multiple connection) 52
- Timeout (file transfer in synchronous emulation) 80
- TN3270 48
 - Host name 48
 - Port 48
 - Terminal type 48
- TN3270E 40, 48, 104
 - Host name 48
 - Logical unit name 49
 - LU 40
 - Port 48

- Terminal type 48
- TN5250 49
 - Host name 50
 - Port 50
 - Terminal type 49
- TN5250E 40, 105
- Toolbar 17, 45
 - Customization 17
 - Docking 17, 45
 - Style 17, 45
- Transparent printing 157
- Transparent text (hotspots) 113
 - Events 119

U

- Underline color 166
- UNISYS 15
- UNIVEL 15
- UNIX SNA_TCP/IP gateways
 - Mode TN3270/TN3270E 40
 - TN5250 Mode 43

W

- Wild-card characters (hotspots) 115
- Wildcards (hotspots) 115
- Winsock 14
- WYSE 15

X

- X-Modem (File transfer protocol) 79
- Xany/Xoff 20
- Xon/Xoff 20

Y

- Y-Modem (File transfer protocol) 79

Z

- Z-Modem (File transfer protocol) 79