ESKER *Tun*® *Plus*

PC and Multi-User
Administrator Guide

ESKER®
Document Process Automation

Tun Plus 2009
Issued May 2008

Esker S.A., 10 rue des Émeraudes, 69006 Lyon, France
Tel: +33 (0)4.72.83.46.46 ♦ Fax: +33 (0)4.72.83.46.40 ♦ info@esker.fr ♦ www.esker.fr

Esker, Inc., 1212 Deming Way, Suite 350, Madison, WI 53717 USA
Tel: +1.608.828.6000 ♦ Fax: +1.608.828.6001 ♦ info@esker.com ♦ www.esker.com

Esker Australia Pty Ltd. (Lane Cove - NSW) ♦ Tel: +61 (0)2 8596 5100 ♦ info@esker.com.au ♦ www.esker.com.au

Esker GmbH (München) ♦ Tel: +49 (0) 89 700 887 0 ♦ info@esker.de ♦ www.esker.de

Esker Italia SRL (Milano) ♦ Tel: +39 02 57 77 39 1 ♦ info@esker.it ♦ www.esker.it

Esker Ibérica, S.L. (Madrid) ♦ Tel: +34 91 552 9265 ♦ info@esker.es ♦ www.esker.es

Esker UK Ltd. (Derby) ♦ Tel: +44 1332 54 8181 ♦ info@esker.co.uk ♦ www.esker.co.uk

Table of Contents

Connecting a PC to a network

Connecting SNA networks

Due to the way the Internet's developed, the current trend favors the TCP/IP protocol over other protocols. As a result, SNA networks have had to accommodate TCP/IP solutions. This migration currently uses a gateway between the TCP/IP and SNA networks. This type of gateway lets a TCP/IP network communicate with IBM MainFrames or AS/400 machines in an SNA environment.
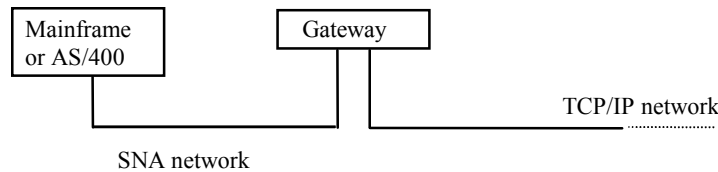
Gateways with 3270

Esker's 3270 emulator can connect to MainFrame systems in three modes:

• Directly to the MainFrame server using the TN3270 and TN3270E protocols. TN3270 is the communication protocol used in synchronous emulation in the IBM mainframe environment. The TN3270E protocol is an extended version of the TN3270 protocol.

• Via a UNIX SNA-TCP/IP gateway in TN3270 and TN3270E mode

• Via the Microsoft SNA-Server and Novell Netware for SAA gateways in native mode.

In TN3270 (or TN3270E) mode, the PC emulates a 3270 (or 3270E) terminal by connecting to the gateway as if it were a computer.

In native mode, the emulator communicates with a gateway-specific DLL on the PC. The DLL then handles the connection with the gateway.

Before you use the 3270 or 5250 emulator for emulation via a Microsoft SNA Server gateway, first install the client side of SNA server (SNA Server Client, version 3.0), supplied with Microsoft® BackOffice®, on your PC. Refer to Microsoft's documentation for installing this product.

Novell's Netware client and Novell's IntranetWare for SAA client must be installed on the PC on which the emulator is run.

Gateways with 5250

Esker's 5250 emulator can connect to AS/400 systems in three modes:

• Directly: Using the TN5250 or TN5250E protocol over a TCP/IP network. TN5250 is the communication protocol used in synchronous emulation in the IBM AS/400 environment. The TN5250E protocol is an extended version of the TN5250 protocol.

• Using a UNIX SNA-TCP/IP gateway in TN5250 and TN5250E mode.

• Using Microsoft SNA-Server and Novell Netware for SAA gateways in native mode.

Standard APPC APIs (Application Program-to-Program Communication) are used to connect to the gateways.

Connecting via a gateway in IBM emulation

▶ **ActiveX version**

In Esker Viewer, select the New option in the File menu to open an IBM emulation session (3270 or 5250), and then select the connection type corresponding to the gateway used

In an emulation session previously opened in Esker Viewer, select the Connection option and then Configuration in the Session menu, then select the connection type corresponding to the gateway used.

▶ **Executable version**

In the emulator executable version, select the New session option in the File menu, and then select the connection type corresponding to the gateway used.

4

Terminal Emulation

Creating, using, and customizing emulation sessions are described in detail in the Application Access Guides. There are three such guides included with Tun Plus: one for 16-bit executable sessions, one for 32-bit executable sessions, and one for ActiveX sessions. This section provides an overview of Tun's emulation features.

Asynchronous connection

These types of connection are used by most standard UNIX terminals (SCO, DEC, WYSE, HP, etc.). Esker's asynchronous emulator supports the following types of connection:

• The PC can establish a Telnet connection over the TCP/IP network.

• The PC can connect to the server using a serial link link over one of the serial ports (COM1 through COM32).

• The PC can connect to the server over a phone line with a modem.

• The PC can connect to the server using a specific protocol like Novell's IPX. These connections use the 14h and 6Bh BIOS interrupts if the network card and the communication software allow it. The interrupts redirect the connection to the serial port of the PC. The redirection is transparent to the user.

The dial-up connection and the serial link enable PCs that aren't on the network to connect to the server with a cable and/or modem. Esker's asynchronous emulator supports this type of connection to allow PCs that aren't directly connected to the company's computing system to access it.

Notes:
Novell supplies a TSR (Terminate and Stay Resident) program, NVT.EXE, that acts as an interface between the NVT/IPX software and the BIOS interrupts. This program must be run in MS-DOS before you can connect to the server. To compensate for this disadvantage, Esker supplies a version of NVT.EXE which opens and closes NVT/IPX protocol connections by toggling between Tun EMUL and a DOS session.

The use of the BIOS 14h and 6Bh interrupts is limited to the 16-bit version of Esker's emulator.

You can change the type of terminal you want to use once the connection with the server is established.

Connection to an IBM MainFrame server (3270)

IBM mainframe servers transmit and receive the 3270 type of data flow.

To communicate with this type of server, the PC can establish the following types of connection:

• Direct connection to the server using the Telnet 3270 (TN3270) protocol through the TCP/IP network . The Telnet 3270 protocol is defined in RFC 1041.

• Direct connection to the server using the extended version of this protocol: TN3270E. The TN3270E protocol is defined in RFC 1647.

• Connection through a UNIX SNA–TCP/IP gateway in TN3270 or TN3270E mode.

• Native mode connection via a Microsoft SNA Server or Novell Netware for SAA gateway.

The TN3270E protocol compensates for some of the shortcomings of the TN3270 protocol. Notably, the extensions are:

- An LU (logical unit) is attributed to each connection: An LU is an IBM-specific element similar to UNIX devices. For example, a printer connected over Telnet to an IBM mainframe always uses the same LU. Additionally, one or more LUs are often associated with a particular user profile. This means you can access a particular application menu according to the LU used. TN3270E emulation lets you choose a specific LU related to the processing you want to do.

- The SYSREQ and ATTN function-keys are supported. The SYSREQ function-key interrupts the current application to return to the system menu. The ATTN function-key interrupts a process at the Telnet level.

- 328x printer emulation.

- Connection to an SNA network through an SNA-Server gateway.

  The native API 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 (3270 emulator) need only send a connection request over the network for it to be answered by whichever of the gateways is available.

  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.

Connection to an AS/400 system (5250)

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:

- Direct connection to the server using the Telnet 5250 protocol through the TCP/IP network (TN5250).

- Direct connection to the server using the extended version of this protocol: TN5250E. The TN5250E protocol is defined in RFC 1205.

- Connection in TN5250 or TN5250E mode through a UNIX SNA–TCP/IP gateway.

- Connection in native mode through a Microsoft SNA Server or Novell Netware for SAA gateway over TCP/IP or another type of network.

  The TN5250E protocol compensates for some of the shortcomings of the TN5250 protocol. Notably, the extensions are:

- 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.

Customizing terminal emulation

The following customizations are described in detail in the appropriate Application Access manual (ActiveX, 32-bit or 16-bit Executable) depending on the type of emulator employed. This section provides an overview of the features available to you through Esker emulators.

Changing the display font

A typical terminal display always contains a set number of characters (24 or 25 lines by 80 or 132 columns). On this type of terminal, the font used is called a "fixed" (non proportional) font as all the characters use the same amount of space on the screen.

On an emulation display, you can use any of the Windows fonts as long as it isn't proportional. Additionally, to view the semi-graphic characters, you must use an OEM and not an ASCII font.

Esker's asynchronous and synchronous terminal emulators provide a fixed OEM font in addition to the normal fonts. The font comes in different sizes for 80- or 132- column displays and is designed for the most convenient use of the emulator (80-column SystemPC font and 132-column Sys132PC font).

You can choose the font and font size from the terminal emulator's font list, regardless of the width of the display (80 or 132 columns).

Additionally, you can select the Dynamic Sizing option to adopt the size of the font to the size of the emulation window: That way, the emulator always displays a full terminal screen. This option is provided by the system in synchronous emulation (3270 and 5250).

Resizing the emulation screen (asynchronous emulation)

Most of the emulated terminals are 25 lines by 80 columns. As these dimensions are the most common, it usually isn't necessary to change the settings for the emulation window dimensions.

For some terminals (132 columns or more than 25 lines), however, you can resize your screen to suit the dimensions of the emulated terminal.

If you choose to display 80 columns in 132-column emulation, you can use the horizontal scrollbar to view the other part of the screen.

Increasing the number of emulation screen lines stored

The physical limits of the emulation window are sometimes insufficient to interpret all the information coming from the server. You might find it useful to store more lines than the number displayed in the emulator.

You can store up to 1024 lines and access them using the vertical scrollbar.

Reformatting displayed characters

The characters displayed on the PC screen during emulation are those typically displayed on any emulation screen. Tun emulators apply revamping techniques that embellish traditional attributes with enhanced graphics, as becomes the Windows environment.

You can choose style and color attributes for the characters displayed:

• Foreground color (text) and background color.

• Character style (italic, bold, underline).

• Text style (normal, raised, inset, shadowed).

• Border style (indent or outdent borders, indent, merged borders).

Two, three or four attributes can be changed to produce all the possible combinations. Example: You can display underlined text in blue, highlighted text in red and highlighted underlined text in black.

In synchronous emulation (3270 or 5250), the attributes are different but the same enhancements can be made to the color and style of each attribute.

Displaying the attribute characters (3270 and 5250)

Character strings may include special characters that contain information on the characters that follow (for example, on character string format). Application developers sometimes need to see these characters on the screen to design their screens.

The Esker synchronous emulators can display these special characters on the screen and assign them specific colors to distinguish or hide them.

Customizing the screen background

You can replace the monochrome background with a bitmap image. You can make these changes for all the emulation types (asynchronous, 3270 and 5250 emulation).

Example of a standard UNIX application's output : Without a background:

Using function-key panels

A function-key panel is a set of buttons on a graphic keypad. Each button has its own function to fulfill. One or several actions predefined in the function-key panel can be triggered by a single mouse click on a button. For example you can use the key panel to replace the keyboard keys most frequently used in an application.

Esker's emulators are supplied with a function-key panel editor so you can design your own function-key panels with text and bitmaps. Each button triggers a particular action: Send a character or string of characters, execute an action like a file transfer, start an application, run a macro, etc.).

Display controls on a screen

You can improve a traditional application by adding controls, in other words buttons, menus, customized text, combined lists, scroll bars, that will be automatically placed on the screen when a particular character string is displayed. Each control may be associated with one or several actions activated either by the user, or in some cases as soon as a searched character string is recognized.

For example, a button may be added to an emulation screen so that the user can start a printout by simply clicking on this button. Or a file transfer may automatically occur without any user intervention when a given application screen is displayed.

Identification of the string and installation of controls are two successive operations managed by the hotspots function

Assigning a special action to a key

Each key on the keyboard is programmed in the emulator to send a specific code, which usually corresponds to sending a character (alphanumeric key), but sometimes to execution of an action (send a character string, a function key, execute a script, a specific action, etc.).

The default settings proposed by Esker are usually sufficient for typical use of emulators. However, Esker terminal emulators do have a keyboard editor that you can use to assign actions of your choice to each key. You can thus modify the default configuration, for example if you have a very specific terminal emulation or a specific linguistic environment.

Working with simultaneous sessions

You'll find it useful to be able to access different centralized applications (accountancy, sales, stock control applications, etc.) at the same time from an emulation session. To do that, you open several sessions one after the other. This also lets you see different parts of an application on different emulation screens.

In active X version

Esker Viewer contains a number of ActiveX components that can be instantiated as many times as necessary (MDI (Multiple Document Interface) technology). You can thus start several types of session (UNIX asynchronous emulation, 3270 emulation, FTP session, etc.) and several occurrences of the same type.
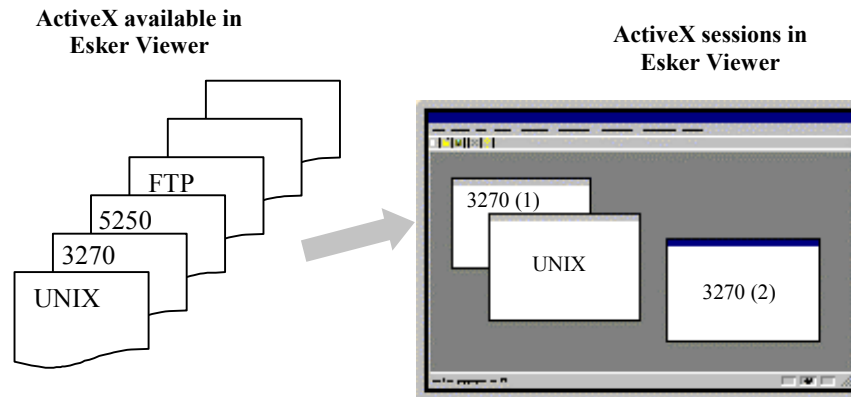
Use the Window menu to change from one session to another.

In executable version

The asynchronous emulator Emul is multi-session: You can open several sessions on different servers, using different communication modes, if necessary (for example, you might run a TCP/IP session at the same time as an RS-232 session). You can also open multiple simultaneous TCP/IP sessions with the same server.

You can switch from one session to another by using the keyboard or function-key panel buttons you've created to do that. For most emulations you can use the keys Alt-F1 through Alt-F4 to switch sessions.

Displaying characters on the PC screen as they look on a terminal

Servers and PCs don't always code characters the same way, depending on the UNIX environment. The relationship between the character sent and the character displayed differs from one environment to another, hence, the existence of character tables.

Example:
To code an "e", the ISO8859 character set uses the code e9, while the CP850 character set uses the code 82. If the server uses ISO8859 encoding and the PC uses CP850 encoding, the terminal emulator must handle the conversion between the two tables.
The terminal emulator therefore always uses one or more translation tables, depending on the type of emulation. This conversion is even sometimes applied to a single emulation type that's used with two different versions of UNIX (this is the case with IBM 3151 emulation which requires one of two tables depending on the AIX version used).

Each type of emulation that Esker provides includes character tables as standard to ensure correct emulator screen display. If you've problems with the display, you should check the character tables used by your UNIX system.

Printing from an emulation session

Esker emulators offer several print modes, that may be used as a function of requirements:

• Conventional hard copy of an emulation screen: you choose to print the contents of the emulation window, in text or graphic mode. You can print on any printer declared on the PC (local printer, network printer through LPR).

• Printing with template: this mode is provided for synchronous terminal emulations (3270 and 5250). You can use it to define a print model to print a predetermined set of pages.

• Transparent printing: this mode is provided for the asynchronous terminal emulation (UNIX, HP, DEC, etc.). Using actions provided by Esker, it can redirect data flows to a printer connected to the PC rather than to the screen for dis-

play.

- Printing to a 3287 or 3812 printer: this mode prints data from an IBM MainFrame (Telnet 3270 protocol) to a 3287 printer, or data from an AS/400 minicomputer (Telnet 5250 protocol) to a 3812 printer.

Transferring data in an emulation session

Protocols like FTP, TFTP, NFS, etc. are used to transfer files to and from a PC. When the user is working from a terminal emulation, he can make use of the ActiveX FTP file transfer component, within Esker Viewer. He can also remain within the same emulation session to make a data transfer, using one of the various protocols provided by Esker.

Esker includes a file transfer function in the asynchronous emulator and the 3270 synchronous emulator. There are two possibilities: either the server initiates the transfer operation by sending an escape sequence, or you do this yourself by sending specific commands to the server.

The asynchronous terminal emulator supports the ASCII transfer protocol, the standard X-Modem, Y-Modem, and Z-Modem transfer protocols used by BBSs (Bulletin Board Systems), and also uses the proprietary transfer protocol RTUNPLUS.

The 3270 synchronous terminal emulator can transfer files through IND$FILE. It offers a multiple file transfer option for sending or receiving a set of files.

Macros

Terminal emulators provided by Esker are capable of writing and executing macros in order to automate some tasks. For example, the different stages of starting an application or disconnecting can be made transparent (login and password request, starting an application, etc.).

Different data formats are used for automation by macros, depending on your emulator version (ActiveX or executable):

- In the ActiveX version, macros are Jscript or VBScript functions grouped in macro libraries (files with the .js or .vbs extension.

- In the executable version, macros are written in the Esker proprietary language (EScript) and are recorded in files with the .mac. extension

---

Note:
Macros written using the Esker proprietary language are fully compatible with the ActiveX version of emulators (files with the .mac extension).

The JScript and VBScript script languages can also be used to write small simple scripts directly associated with an emulation element (a button, menu, mouse event, keyboard key) without the need for a file. These scripts are written using a simple script editor included in Esker Viewer.

Writing macros in the ActiveX version

Before you can write macros for the ActiveX version of emulators, you need to know:

- A script language compatible with the ActiveX Scripting standard (JScript or VBScript).

- The emulation ActiveX APIs documented in the emulapi.hlp file. This file is available in the \Docs\Misc directory of the Tun PLUS installation CD-Rom.

Esker Viewer provides a tool for the management of macros that includes a macro recorder, and can be used to edit, create or modify macros.

Writing macros in executable version

Before you can write macros for the executable version of emulators, you need to know the EScript language developed by Esker and documented in the escript.hlp file. This file is available in the \Docs\Misc directory of the Tun PLUS installation CD-Rom.

To create a macro in the asynchronous emulation (UNIX), you first use the macro recorder: This records what you type and translates it into simple Send (character emission) and Sleep (time-out) instructions. You can then edit the macro (with Notepad, for example) and improve it by adding other instructions.

Executing macros

You can execute a macro at anytime during your session by starting to execute it manually. You can also associate a macro with a keyboard key, a mouse event, or a hotspot.

The advantage of a macro is that it can automate some tasks. Connection and execution of given application are two tasks that users frequently want to automate. You can do this by associating a macro with a session in the same way as you associate a terminal (in other words all emulation files) and a context (colors, screen background, key panel, etc.) with it. The macro may be associated with the start of the session (in the case of a connection macro), or with the end of the session (and in this case the macro will be executed when the session is closed).

Running a Windows application from an emulation session

Terminal emulation on a Windows PC lets you run UNIX and Windows applications at the same time. This is useful when you want to copy data from the UNIX system to a Windows application document, as, for example, from a UNIX business application to an Excel spreadsheet.

To run a Windows application from an emulation session, you can do any of the following:

• Create a button to this effect in a function-key panel.

• Create a hotkey for the task.

• Associate a control (hotspot) with this function, and display it on the screen after a string has been recognized.

• Attribute an escape sequence to the task and include it in an application menu.

• Link the operation to a mouse event.

A function-key panel is a fully customizable tool for which you can create buttons that perform individual tasks. For example, a button can send a specific character string or execute a macro.

DDE (Dynamic Data Exchange)

A DDE server (Dynamic Data Exchange) like Esker's asynchronous emulator lets a Windows application access the emulation screen. The application can then exploit server data dynamically through the terminal emulation session and the link established between the two environments (the Windows application and the UNIX emulation screen).

In practical terms, you use DDE when you copy a part of the emulation screen (Copy option in the Edit menu) and paste it (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.

HLLAPI (High Level Language Application Programming Interface)

When IBM produced its 3270 emulator (PC3270), it defined a comprehensive set of functions that allows third-party applications to control the emulator. These functions, collectively known as HLLAPI (High Level Language Application Programming Interface), let a Windows application (written, for example, in C, C++ or Visual Basic) retrieve

data from an emulation session without having to display the emulation screen. Communication with the server is transparent to the user.

Unlike DDE, HLLAPI requires the application to call DLL functions that comply with one of these standards:

• IBM's EHLLAPI (E for Emulator)

• Microsoft's WinHLLAPI

Esker's 3270 synchronous emulator supports both these standards.

Workspaces and configurations

Tun permits user to create and custom work environments, which may include multiple emulation sessions or other resoues. These are described in detail in the appropriate Application Access manual (ActiveX, 32-bit or 16-bit Executable) depending on the type of emulator employed. This section provides an overview of their features.

ActiveX workspace

In Esker Viewer, a workspace corresponds to the set of files and parameters necessary to setup one or several sessions: number and types of sessions to be opened, session opening and closing parameters, display windows, menus, tool bars, etc.

Parameters related to workspace session(s) are stored in a file with the .cwz extension.

Note:
A workspace is used to save parameters for one or several ActiveX sessions accommodated by Esker Viewer, within a single file. For upwards compatibility reasons, the concepts of IBM sessions (.cfs and .ini files), UNIX configurations (.cfg and .cfz files), and macros (.mac files) used in previous versions of Tun, have all been kept; you can load files of this type and then integrate them in a workspace, you can run a macro in the old format, you can save individual sessions in the .cfs or .ini format for IBM, or the .cfg or .cfz formats for UNIX. Note that the only way of saving more than one session now is in a workspace (.cwz file).

Executable configuration

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, hot spots) and a TCP/IP session on an hpterm terminal with display settings B.

Esker's asynchronous emulator can group all settings files for the session or sessions (terminal, keyboard, escape sequence files, etc., macro and key panel files, etc.) in an archive (file with .cfz extension). Once created, the archive can be used to distribute settings files for shared use between several users. For example, this distribution may be made by diskette, on a NIS server, via TFTP, etc.

For upwards compatibility reasons, you can load configurations with the .cfg extension in the asynchronous emulator. In this file type, parameters related to the configuration session or sessions are simply memorized within a file with a .cfg extension . This file indicates which files are used for the configuration session(s). However, you will only be able to save a new configuration in a .cfg archive file.

In Esker's IBM synchronous emulator, the settings for the configuration session(s) are stored in a file with the extension .cfs (only one session possible for the synchronous emulator).

Defining settings for the emulator work environment

ActiveX version of emulators

Apart from defining settings specific to each session (for example redefinition of colors, association of a macro, a key panel, etc.), you can also define settings for your work environment:

• Customization of menus and tool bars

• Choice of Esker Viewer start and exit options, file backup and open options.

Executable version of emulators

You can customize your working environment by defining the emulator startup and session settings. That way, you find the same visual display each time you start the emulator or open a session.

The startup options let you:

• Choose the startup screen size: This can be the maximum size, the size of the screen in the previous session, reduction to icon form, the default size set in a different dialog box, or the full screen (without menus).

• Choose save options on exit from the emulator.

• Choose the default screen size for the opening of a new session (maximum size or default size, asynchronous emulator) .

• Choose the language of your physical keyboard (asynchronous emulator).

• Choose the size of the customizable toolbar (asynchronous emulator).

Distributing programs

You can redistribute Esker emulators in many ways:

• You can reuse the same work environment on different occasions without needing to redefine your own parameters each time. For example, you always use a 3270 emulation session and a 3287 emulation session for which you have set some parameters.

• You can redistribute this type of work environment by making it independent and therefore usable on systems other than your own. For example, you have prepared a UNIX session and included its parameter files in an archive, and you supply this archive to other users with execution programs.

Finally, you can redistribute a new executable that you customized and generated yourself, to supply it to other users. For example, you have configured a set of UNIX sessions and chose the menu bars and tools for the executable. You thus supply a "turnkey" product to other users.

5

Sharing Network Resources

Creating, using, and customizing network resources are described in detail in the Network Resource Access Guide. This section provides an overview of Tun's networking features.

NIS

To facilitate network management, Tun uses the NIS standard (Network Information Service) in the form of an NIS resource browser.

The principal of NIS is as follows: A UNIX server manages network resources stored as tables.

Typically, the NIS server is used in a master/slave architecture: The master server manages the NIS domain, and one or more slave servers store duplicate tables for the NIS domain. Each client of the NIS server can call any of the servers.

NIS Domain

Master NIS Server

Slave NIS Server

NIS client 1

NIS client 2

NIS client 3
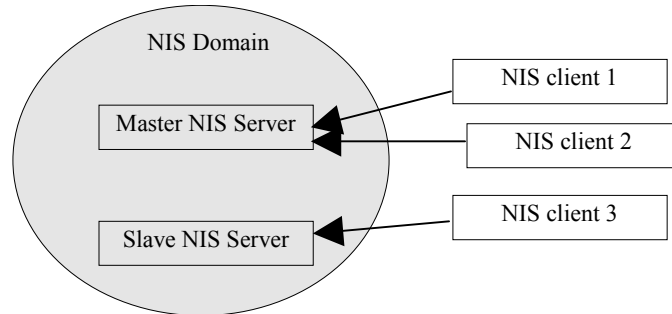
The NIS Browser allows:

• The administrator to manage the NIS server resource tables from a Windows machine; he can create, update, configure and export (make available) resources.

• Users to access resources directly without any previous knowledge of resource configuration; with a simple mouse click, the user can mount a network drive, install a printer, start an application, open an emulation session, etc.

Printing from a PC

Depending on the print job and your work environment, data transfer between two machines can take different forms:

• From a terminal emulation session: You can use transparent printing, or simply make copies of the screen.

• 3287 printer: You can send 3270 data streams from an IBM MainFrame server to a printer connected to a PC.

• From a 3270 emulation session: You can print screen pages using printing templates.

• LPR printer redirection and LPD printer sharing: From your PC, you can print to a remote printer connected to the network. You can also share your printer with the rest of the network by making your PC an LPD print server.

NFS

To access files and directories on a different network machine, you can use the NFS solution (Network File System). This protocol lets a PC user use remote drives as if they were local resources, and also export (make available) his own resources for other users. This can be a temporary or a permanent arrangement.

A PC can act as an NFS client: You access a part of a remote drive exported by an NFS server. Depending on your access rights, you can read and write data, and run programs.

A PC can also act as an NFS server: In this case, you export all or part of your local drive. Remote users see your exported disk space as part of their local environment. The NFSD protocol, the server side of the NFS protocol, is used in this case.

NFS is a transparent connection that's independent of the operating system. The NFS protocol was conceived and developed by SUN Microsystems. It was later standardized by X/Open and extended to include PCs under the name (PC)NFS.

The originality and strength of NFS lie in the fact that the server is "stateless". This means it doesn't store a specific context for each client. When a client asks to read a part of a remote file, the name, position and size of the part to read are transmitted to the server. The server then opens the file, positions itself on the section to read, reads the section, returns the results and closes the file. When the transaction is finished, the server "forgets" the client.

The advantage of this solution is the absence of a permanent connection between the server and the client. If the client reboots, for example, it has no effect on the server (there are no residual processes on the server). If the server stops, the client only loses the use of the remote drive while the server's not functioning.

Security with NFS depends on normal UNIX user management. Typically, file and record locking are supported by NFS clients and servers, as is the case with Tun NFS.

Tun NFS gives users the following benefits:

• File sharing and exchange between multiple users.

• Centralized installation and backup of programs and files, accessible from any PC on the network.

• NFS V3 protocol compatibility in Windows 98, 2000 and XP: A read cache is used to give improved performance.

SMB

SMB (Server Message Block) is a file-sharing protocol installed on a server that lets network clients see the server's file system as if it were part of the normal Windows environment.

The main advantage of SMB is its centralized approach: You only need to install it on the server and configure the server correctly to make the server's drives and resources available to network clients. No administration is necessary on the PC.

Transferring data between two machines (PCs or servers)

In addition to terminal emulation data transfers (mentioned in the previous section, data transfer between two machines can take different forms depending on your intentions and your work environment:

• Using the FTP protocol: FTP was largely developed through the Internet. You can use FTP for file transfer and to transform your PC into a file server.

• Using TFTP: TFTP (Trivial File Transfer Protocol) is a simple protocol that lacks authentication procedures. You can only read and write data on a server that authorizes it (not available in the Citrix/Microsoft TSE version of Tun).

• Using a macro to automate the transfer: You can exploit Tun EMUL's or Tun FTP's macro languages.

• Backing up and restoring data: Use Tun TAR for TAR format backups on remote devices.

Transfers between FTP client on PC and an FTP server

You can transfer files between a PC and an FTP server using the FTP protocol (standard UNIX servers, IBM/MVS servers, AS400 servers, PC server). The FTP protocol ensures the security of transferred data with login ID and password procedures for each connection established with the server.

The transfer mode is either binary (no data conversion) or ASCII (carriage returns and line feeds handled).

The FTP server may also be a PC running an FTPD server program.

In addition to the basic FTP protocol which allows file downloading and uploading, Tun FTP offers the following advantages:

• File and directory transfer using drag and drop in a 100% Windows environment.

• File conversion with filters (binary or ASCII transfer mode). These filters convert Windows files with carriage return characters into UNIX text files with line feed (LF) and no carriage returns (CR). They also convert PC format accented characters into UNIX format (and vice-versa).

• Multi-session implementation allowing transfers between two servers without using temporary files on the PC.

• Session automation using the built-in macro language.

Transfers between two FTP servers

You can transfer data between two UNIX FTP servers from a PC using the FTP protocol. There's no need to create temporary files on the PC. You simply connect via the PC client to the relevant servers and drag and drop the files or directories from one server window to the other.

If the two servers are in the same type of environment, you can transfer the files in binary mode: You don't have to worry about line feeds as with Windows PC - UNIX server transfers.Trivial file transfer

You can transfer data quickly without login ID and password security using the TFTP protocol (Trivial File Transfer Protocol). This protocol simplifies file transfer (with the advantage of speed) by authorizing, for example, a transparent file transfer at bootup (useful for loading configuration files). The transfer mode is binary. Reads and writes to a TFTPD server are allowed.

Automating FTP data transfer

With FTP client, you can automate data transfer by using macro program instructions. These instructions are part of the macro language included with Tun FTP. They can be used to control file transfer without keyboard or mouse input.

Backing up and restoring data

You can back data up on a backup device attached to a server (and restore it) using the TAR format. Since, the PC rarely has direct access to a large enough backup device, the UNIX TAR backups give Windows PC users access to mass backup storage.

To perform file backups and restores, Tun TAR establishes an RSH or REXEC (password required) connection with the UNIX machine and sends a write request (for a backup) or a read request (for a restore) to the backup device.

Sharing a printer with the rest of the network

When a printer is connected to a PC and isn't configured for network use, only the PC can use it. The PC can act as a "relay" between the printer and the rest of the network (UNIX machines, other PCs), allowing the printer to be used more widely. To do that, the PC must act as a print server.

A print server uses the LPD or RSH protocol to enable other network machines to print via TCP/IP on the printers it controls. LPD works as follows:

• The PC connected to the printer you want to share runs a print server program (LPD or RSH protocol).

• When a UNIX machine wants to print to a shared printer, it establishes an LPD or RSH connection with the PC server using the logical name assigned to the PC. A PC can do the same thing if it also has the remote printing client program (LPR protocol).

• The characters to print are sent over the established connection, then transmitted by the print server software to the appropriate printer port.

Notes:
This type of connection between the UNIX machine or the PC client and the PC server has the advantage of not being permanent. If one side or the other stops functioning, no traces of the connection are left behind.
Even if the UNIX machine doesn't have LPD to send the data to the printer, it can use the RSH client protocol.

To configure the remote printer:

• Assign a logical name to the printer. This name is used by the UNIX machines and PCs that want to print to the printer.

• Declare the printer's real name on the PC's print server.

• Set the required time-out and UNIX/Windows parameters.

Each system (PC clients, UNIX machines) can then send print jobs to the configured printer.

Note:
The remote printing command used on the UNIX machine depends on the system and protocol installed. Refer to the user manuals of the machine and see the section "Using public printers with RSH", chapter "Sharing PC Printers" in the Network Resource Access manual.

Tun TAR

After establishing a connection with the server the backup device is attached to, the client application writes the backup data to the device or reads data from the device. On UNIX systems, backup is typically performed using the TAR format.

The two communication protocols used are:

• RSH: Backs data up or restores it without requesting a password. You must, however, configure the UNIX server the backup device is connected to, so that it recognizes the user and his PC.

• REXEC: User must enter a login ID and a password when performing a backup.

Executing remote commands

You can perform simple, everyday operations on the server from a Windows PC without having to run an emulation session. The advantage of using remote RSH and REXEC commands compared to starting an emulation session (Telnet) is that the connection is closed after the execution of the command: Fewer demands are made on network resources.

These simple operations include:

• Query the server on queue status.

• Run a procedure.

• Display the results of the commands on the PC, or copy the results to the Windows clipboard.

Tun RSH uses the RSH or REXEC service to execute remote commands (RSH: No password required but the server must be configured; REXEC: Password access only).

You can also include several commands for the same server in a macro and run the macro from a customized toolbar.

6

Accessing Databases

Use Tun SQL (included as part of Tun Plus) to access, query, and simplify databases. The following features are described in detail in the Data Access Guide. This section provides an overview of Tun's database accessing features.

Installing an SQL server on a UNIX machine

To allow access to a UNIX DBMS, you must install an SQL server program adapted for the machine and the DBMS used.

Esker supplies SQL servers for most platforms and DBMSs on the market.

Verifying the installation of an SQL server on a UNIX machine

Before you can access a UNIX server to query data, you must verify that the SQL server is correctly installed on the UNIX machine. There are two ways to do that:

• On the UNIX machine: Verify there's an SQL server by sending a UNIX command (on the machine itself or from a terminal emulation session).

• On the Windows PC: Use Tun DB Show, a utility supplied by Esker with Tun SQL.

Simplifying a view of a remote database

Esker has included in Tun SQL a module (Tun DB Revamp) that simplifies a user's view of a database. Tun DB Revamp associates a simplified structure, known as a virtual database, to the real database.

A virtual database only contains the information required by the users for whom it was created. The presentation of the information is customized to suit their needs. A "virtual" ODBC driver is used to query the virtual database: This driver is, in fact, an interface between the virtual database and the ODBC driver of the real database. The virtual

ODBC driver doesn't query the real database directly. It converts the queries to the virtual database into queries the real database can recognize. It then transmits the converted queries to the real ODBC driver.
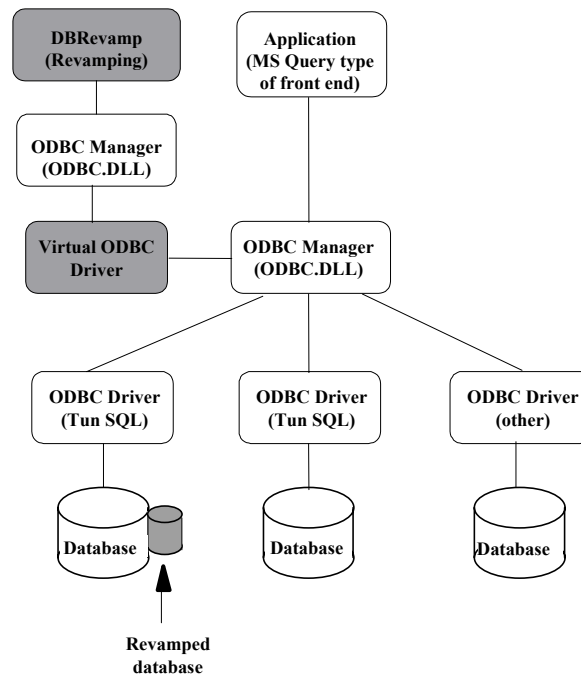
To query a virtual database, you need:

• A virtual data source: This is a link between the real data source and the data tables derived from the real database.

• A virtual ODBC driver: This is a link between the virtual data tables and the ODBC driver associated with the real database.

The advantage of Tun SQL is that the virtual ODBC database allows transparent access to real databases from a "revamped" view of the database's tables and fields.

Querying a remote database without knowing SQL

Data querying tools (like MS Query) can be used in the Windows environment: They allow users to process data without prior knowledge of SQL.

To use such tools you must first assemble the information required for the job in a data source. A data source contains:

• UNIX server identification (name or IP address).

• The type of DBMS used (Oracle, Informix, Sybase, DB2, Progress).

• Database identification.

• Optional information.

Query tools like MS Query use the data source references to perform the user's queries. The graphical interface lets users query the database without necessarily knowing the query language.

With Esker's virtual ODBC driver, you query a virtual database with a tool like MS Query in the same way as you query an ordinary database. Queries are made to the virtual data source, which is the link between the simplified view of the database (known as an "environment", that is, the virtual tables derived from the real tables) and the original data source (UNIX server, DBMS type, database, etc.).

Querying a database from Tun DB Revamp

To facilitate the construction of virtual data tables from real tables, the revamping tool Tun DB Revamp lets you make simple SQL queries to the tables and fields of the real database and the virtual database under construction.

Creating or updating a remote database

You can use Tun DB Script (supplied with Tun SQL) to create or update a database with SQL.

Tun DB Script lets you write an SQL script and submit it to an existing database. The principle is simple: Using a data source, you pass SQL instructions by executing the script.

Index