AT&T

AT&T 3270 Emulator +
ESCORT™

User's Guide

AT&T

308-380
Issue 1

AT&T 3270 Emulator + ESCORT™

User's Guide

NOTICE

The information in this document is subject to change without notice. AT&T assumes no responsibility for any errors that may appear in this document.

Contents

1. Getting Started

1 Getting Started

Overview

This user's guide contains the information you need to know to use ESCORT™.

This guide assumes that a user knows how to

□ operate an IBM® 3278 or DEC® VT100™ (or similar) terminal

□ use an application running on a synchronous or asynchronous host (for example, an IBM host system under IMS, TSO, VM/CMS, CICS™, or VTAM)

□ use an application running on a UNIX® operating system-based AT&T 3B processor.

Before you begin using ESCORT, you should understand

□ the modes of operation featured in ESCORT

□ the special features of the software

□ the concept of accessing multiple sessions.

What Is In This Guide

Organization of This Guide

This guide is divided into five parts:

Getting Started

contains information about this user's guide. Read this chapter to learn how the guide is organized, and what conventions are used throughout the guide.

Using ESCORT

discusses the procedures for using ESCORT and accessing multiple sessions. Special features, the types of ESCORT utility files available, and ESCORT special key combinations are also presented in this chapter.

Appendices

Appendix A: Key Sequences

consists of tables that present functions and the associated keys and/or key sequences for specific terminal types.

Appendix B: Environment Variables

provides information on setting environment variables.

Appendix C: Interpretation of Attribute Bytes

contains information on reading attribute bytes.

Glossary

contains definitions for terms and acronyms used in this guide.

Index

lists page references for locating specific items in this guide.

Conventions

Documentation Conventions

The conventions listed below are used throughout this guide:

- Special function keys on your terminal keyboard are enclosed in a rectangle with rounded corners; for example, ESC.
- Standard alphabetic and numeric keys on your terminal keyboard are printed in bold; for example, **f**.
- Two or more keys separated by spaces indicate that you should press each key sequentially; for example, ESC **f 1**.
- Two keys separated by a hyphen indicate that you should hold down the first key while simultaneously pressing the second key; for example, CTRL – **d**.
- Commands, functions, and keyword operands are printed in bold capital letters. Functions always start with a dollar sign ($). For example, $SCAN is a function.
- ESCORT specific key functions and other key functions are printed in capital letters; for example, CLEAR.
- The following type is used to indicate data that the user types at the terminal:

  ```
  escort script_name
  ```

- The following type is used to indicate information that the system displays on the screen:

  ```
  auto script generation started
  ```

- The following type is used to indicate program text:

  ```
  CONNECT (H1)
  ```

- Multi-word operands are separated by an underscore. For example, *str_expr* represents the words, string expression.
- Brackets [ ] indicate optional operands.
- Braces { } indicate a choice of operands.
- The UNIX file path names are shown with the standard slash character (/). Scripts are portable between the UNIX operating system and the MS-DOS® operating system versions of ESCORT and you may, therefore, substitute the

standard UNIX operating system slash character with the MS-DOS operating system back-slash (\) file name separation character.

Note

Throughout this guide, default key combinations are shown for ESCORT specific functions, for example, the ESCORT Interrupt/Resume (I/R) key combination is shown as the key sequence ESC **f 2**. These default key combinations may be amended by the System Administrator for your particular environment.

> **If the ESCORT default key combinations are amended you must substitute the amended key combinations for the default combinations shown in the document text, the example scripts and the sample programs in this user's guide.**

Data Entry Conventions

□ ESCORT is *case insensitive*, which means that it treats lowercase characters the same as uppercase. The exception to this is string constants.

□ The UNIX shell is *case sensitive*, which means that it treats lowercase characters differently from uppercase. Thus, when you invoke ESCORT with parameters from the UNIX shell command line, the parameters, such as file names, must be exactly the same as those in the file system.

Definitions

The following term is used throughout this user's guide.

Host A host session refers to either a synchronous session or an asynchronous session, unless otherwise specifically noted.

Before You Begin

Before you begin using ESCORT, you should review your specific operating procedures with your System Administrator and familiarize yourself with the key sequences used in ESCORT:

□ Your System Administrator will install the ESCORT software on the 3B computer and will arrange for the customization of ESCORT to suit your particular location and operating environment. When you review with your System Administrator the environment in which you will be operating ESCORT, your System Administrator will advise you of any specific restrictions or requirements.

□ You should familiarize yourself with the key sequences applicable to your terminal. Refer to Appendix A or your Quick Reference Card for the appropriate key sequences for your specific ASCII terminal.

Some of the key sequences listed may not be applicable in an asynchronous environment. Refer to the key sequence table for the VT100 terminal, in Appendix A, for the key sequences applicable in an asynchronous environment.

□ The Escape key, ESC, used with certain other terminal keys, performs special tasks in ESCORT. These special key combinations are defined in this section and are also listed in your Quick Reference Card. Familiarize yourself with these ESCORT specific key combinations.

□ Part of the System Administrator's installation procedure includes the customization of the key sequences for the different types of terminals that will be used in your system. Review with your System Administrator any additional key combinations that have been defined for your operating environment. Your System Administrator will also advise of any changes to the ESCORT specific key combinations.

You can list these customized key sequences in the space provided on your Key Sequence Card.

Refer to Appendix B for information on setting the various ESCORT environment variables.

Once you become familiar with the keyboard layout, you can proceed to the next section, which describes the procedures for entering the three different modes of ESCORT.

Other ESCORT Documentation

This User's Guide is part of the *AT&T 3270 Emulator+ ESCORT* documentation. The entire documentation package includes the following:

- □ **AT&T 3270 Emulator+ ESCORT User's and Programmer's Guides**
  AT&T publication number 308-402.
  This binder contains the following three documents:
  - ESCORT Overview
  - ESCORT User's Guide
  - ESCORT Programmer's Guide
- □ **ESCORT Quick Reference Card** and Key Sequence Card, AT&T publication number 308-389.

2. Using ESCORT

2 Using ESCORT

Overview

The first section of this chapter describe the three operating modes of ESCORT. The next section provides a detailed description of the multiple sessions abilities of the software. Read these sections to understand how to access different host and local sessions either interactively or via script control.

The next two sections of this chapter contain information on utility files and command line structure, and the final section describes the operation of the ESCORT specific key combinations. An understanding of the information in this chapter will allow you to make optimal use of ESCORT.

Using Interactive, Script, and Tutorial Modes

There are three modes of operation in ESCORT: Interactive Mode, Script Mode, and Tutorial Mode.

- □ **Interactive** (or **Terminal Emulation**) **Mode.** This mode provides simultaneous access to multiple host applications. Special ESCORT features are available in this mode.
- □ **Script Mode.** This mode allows you to simulate a user on one of the multiple sessions. While in Script mode, ESCORT accepts data from a script instead of from the keyboard. ESCORT can simulate a user by executing the instructions contained in the script.
- □ **Tutorial Mode.** This mode is similar to Interactive mode, however, it can be used to verify data entered by the operator before sending data to the host.

This section contains the procedures for using these three modes.

Interactive Mode

In Interactive mode, you can work on your application just as if you were entering data on an IBM 3278 synchronous terminal or DEC VT100 asynchronous terminal connected to a host computer.

To enter Interactive mode:

1 Enter one of the following commands at the shell prompt, depending on the session type.

   □ For a synchronous host, type ***escort*** and press (RETURN).

   □ For an asynchronous host, type ***escort async.p*** † and press (RETURN).

   You will see the ESCORT banner screen briefly before you get to your initial application screen.

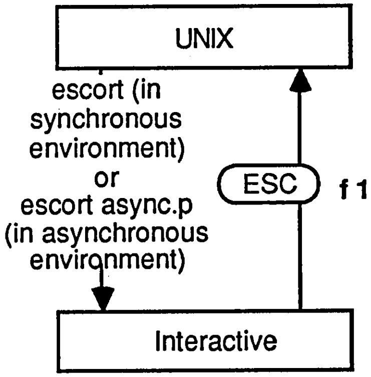
2 Log in to your application and continue to use the terminal as you usually do.

3 To leave Interactive mode and return to the UNIX shell, you can use a special key combination, QUIT. To QUIT Interactive mode, press (ESC) **f 1**.

† The *async.p* ESCORT script is used to initialize the communication port parameters to a specific asynchronous host. The default operands to the SERINIT command contained in the script must be amended to suit your operating environment.

The following diagram illustrates the relationship between the UNIX shell and Interactive mode. You enter Interactive mode on a synchronous host by typing ***escort*** or on an asynchronous host by typing ***escort async.p,*** and return to the UNIX shell by pressing (ESC) **f 1**.

FIGURE 2-1


UNIX
escort (in synchronous environment) or escort async.p (in asynchronous environment)
ESC
f1
Interactive

Caution

As a general rule, you should complete all the required host application operations before quitting Interactive mode. Quitting Interactive mode without completing logoff procedures may not, by itself, log you off the application. The process is dependent on the operating environment as follows:

□ **Synchronous Communication.** When you leave Interactive mode, your terminal session may still be active. This means that if you reenter Interactive mode, you may get back to the same application screen you were at when you executed the QUIT special key combination ((ESC) **f 1**).

The ability to resume an application after executing a QUIT special key combination is dependent upon the protocol being used (bisynchronous communication, *bsc*, or Synchronous Data Link Control in an *sna* environment); the parameter defined in the host system front-end, *VTAM-GEN*; and the application itself. Refer to the appropriate documentation to determine whether you will be able to resume an application in your particular operating environment.

In addition, the controller configuration file in the *sna* environment may affect your ability to resume an application after executing a QUIT special key combination. Refer to the *AT&T 3270 Emulator+ User's and System Administrator's Guides* for further information.

□ **Asynchronous Communication.** Executing a QUIT special key combination in an asynchronous session produces the same result as breaking the host connection by, for example, disconnecting the modem.

If the Interactive asynchronous host operation was executed using the UNIX operating system *nohup* command, quitting Interactive mode will not affect the operation of the application. However, unlike the synchronous environment, you will not be able to resume the application after executing the QUIT special key combination.

While in Interactive mode, you can choose any of the following features of ESCORT:

□ Automatic Script Generation (ASG)

□ Automatic Screen Logging (ASL)

□ Display Screen Attributes (ATTRIB), (synchronous communication only)

□ Local Screen Printing

□ Escape to the UNIX shell

□ Activate, open and close host sessions

□ Display active sessions (SHOW)

These features are described in more detail later in this section.

Script Mode

Script mode is the most powerful feature of ESCORT. In Script mode, you can simulate a user at a terminal. While in this mode, ESCORT accepts data from a script instead of your keyboard. A script is a set of commands contained in a file.

To activate Script mode, type

escort script_name

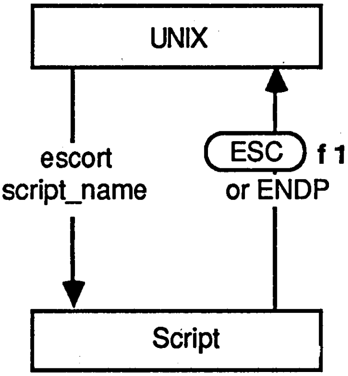
and press RETURN.

The ***script_name*** is the name of the script you want to execute. The ESCORT banner screen appears briefly while ESCORT performs syntax checking and parsing. ESCORT then displays the appropriate host application screen and starts to execute the commands from the parsed list. During execution of the script, data entered on the screen and responses received from the host application may be displayed on the monitor.

When the last statement in the program is reached (the **ENDP** statement), ESCORT is terminated and control is returned to the UNIX operating system. You can also terminate script execution prematurely (before the **ENDP** statement is encountered) by quitting script mode (pressing ESC **f 1**).

The relationship between Script mode and the UNIX shell is illustrated in the diagram below. You execute a script by entering ***escort script_name*** and exit to the UNIX shell with an ENDP statement or by pressing (ESC) **f 1**.

FIGURE 2-2



At any time during active host script execution, you can interrupt script execution, exit to Interactive mode, and then later resume script execution at exactly the same place where you left off earlier.

Use the special key combination, (ESC) **f 2**, to perform an Interrupt/Resume (I/R) operation.

This key combination acts as a toggle switch that alternates between interrupting and resuming script execution. When you interrupt script execution by pressing (ESC) **f 2**, you exit to Interactive mode, not to the UNIX shell.

In the synchronous environment, you may not be able to use Interactive mode immediately. Interactive mode is entered when the current transaction is completed (for example, when the keyboard is unlocked by the application running on the synchronous host system).

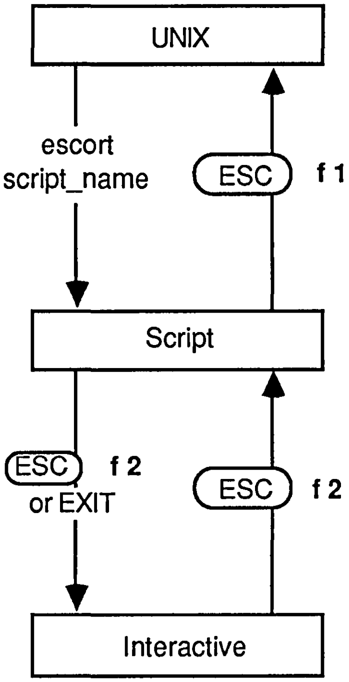
Interactive mode is entered immediately when you interrupt asynchronous host script execution. However, the current transaction may continue to execute.

Another way to enter Interactive mode from active host Script mode is to use an **EXIT** command in a script. The **EXIT** command can be coded anywhere in your program. ESCORT commands are discussed in detail in the *ESCORT Programmer's Guide*.

Once you are in Interactive mode (either due to a manual interrupt by pressing ESC **f 2** or due to an **EXIT** command in a script), you can press ESC **f 2** (the Interrupt/Resume (I/R) key combination) to go back to Script mode.

The following diagram illustrates the relationship of host Script mode to Interactive mode and the UNIX shell. Use ESC **f 1** if you want to quit script execution and return to the UNIX shell. Use ESC **f 2** if you want to interrupt script execution and enter Interactive mode.

FIGURE 2-3



The ability to exit to Interactive mode while executing a script can be used to notify you of an error condition. For instance, if a script cannot handle a particular situation it encounters, the script can inform you of the condition by displaying a message and can then exit to Interactive mode. You can then attempt to recover from the problem condition and resume script execution.

You will automatically be put back into Interactive mode by ESCORT if two error conditions occur. Both of these error conditions cause an error message to be printed in an ESCORT print file and displayed in the operator information area, before exiting to Interactive mode. The two conditions are:

- □ A response has not arrived from the host system before a user-specified time-out occurs. The default time-out period is 60 minutes.
- □ A script attempts to enter data in a protected field on a screen. This situation occurs if you have a bug in a script or if an unexpected screen is received from the host system, in which case the script gets out of synchronization with the application screen.

Tutorial Mode

Tutorial mode provides you with the ability to perform edit checks and validate data entry before sending data to the host for execution. This mode is also useful as an on-line tutorial for your application.

Tutorial mode can be entered from a script by executing one of two **EXIT** commands, depending upon which session is currently connected. Host and local connections are described in more detail later in this section.

If connected to a host session, Tutorial mode is entered by executing

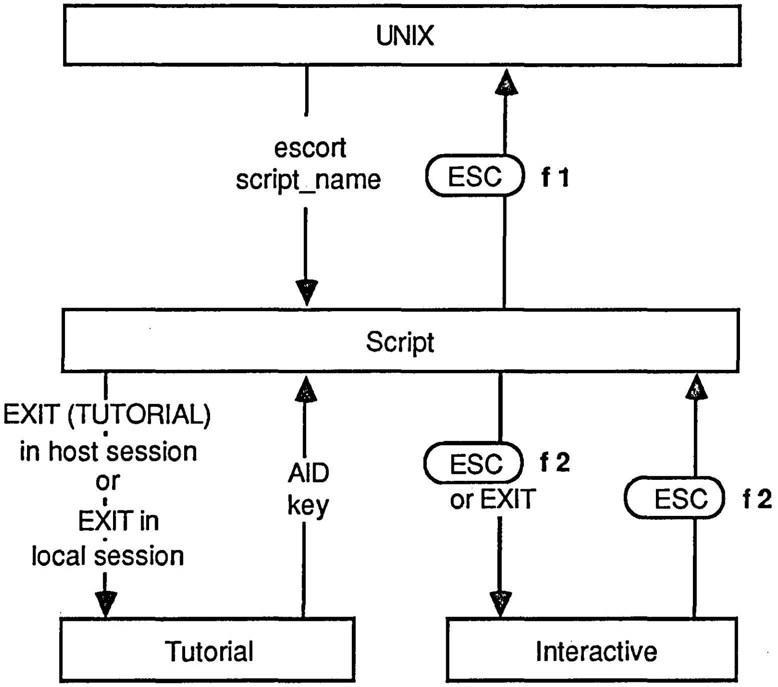
```
EXIT (TUTORIAL)
```

If connected to a local session, Tutorial mode is entered by executing

```
EXIT
```

The following diagram summarizes the means of getting from Script mode to Tutorial mode, Interactive mode, and the UNIX shell.

FIGURE 2-4



Entering Tutorial mode is similar to entering Interactive mode from a script since you enter Tutorial mode by using an **EXIT** command. However, unlike Interactive mode, when you press an Attention Identifier (AID) key (after entering data on the screen), script execution is resumed automatically but the entered data is not sent to the host system.

The AID keys are

□ in the synchronous environment, ENTER, PF1 to PF24, CLEAR, PA1 to PA3, ATTN, and SYS_REQ

- in the asynchronous environment, ENTER, PF1 to PF8 (corresponding to soft function keys F1 to F8) and CLEAR.

In Tutorial mode it is up to the script to decide whether or not to send data to the host. This provides the ability to have a script perform edit checks on entered data and on the AID key pressed. The script may then send the data to the host and use the data as an input parameter later in the script, or reject the data and send a message to you to correct and reenter it.

The way ESCORT handles **AID** commands and AID keys is discussed under the **CONNECT** command in the *ESCORT Programmer's Guide.*

Interactive Mode Special Features

This section describes the additional features of ESCORT that you can use from Interactive mode. They are Automatic Script Generation (ASG), Automatic Screen Logging (ASL), Displaying Screen Attributes, Local Screen Print, and Escape to the UNIX shell.

Automatic Script Generation (ASG)

ESCORT has a feature that automatically generates a script to simulate the actions you perform at a terminal.

To generate a script automatically, you must be in Interactive mode. From Interactive mode, press ESC **f 3**. The message

```
auto script generation started
```

appears at the bottom of the screen. Continue to perform the tasks you wish on your application. After you complete your work, you again press ESC **f 3** to turn off ASG. The following message is displayed:

```
auto script generation stopped
```

Each time you automatically generate a script, it is written to a file, in the directory defined by the ESCDIR environment variable, named *escort.ky{proc-id}*.

To execute the automatically generated script:

1 Exit Interactive mode. Press ESC **f 1**.

2 Execute the script by typing ***escort escort.ky{proc-id}***.

When you run the script, ESCORT automatically performs the same tasks that you just did at your terminal.

Try using this feature to generate a script to perform your login procedures for you. This is a good way to see how ESCORT works for yourself, and you can use this script whenever you need to log in to your application. Sample login and logoff scripts for IMS and TSO are provided on your ESCORT installation diskette.

Note

The generated script may require editing prior to execution so that both variable responses and multiple responses from the host may be anticipated.

Refer to the section, "Synchronizing Data Transmissions" in Chapter 2 in the *ESCORT Programmer's Guide* for further information.

Automatic Screen Logging (ASL)

The automatic screen logging feature saves the image of a specific application screen and any data entered on it. While in Interactive mode in an active host session, you can select the special key combination for Automatic Screen Logging by pressing ESC **f 6**. The message

auto screen logging started

appears at the bottom of your screen. Continue to perform all the tasks you wish on your application. After you complete your work, you again press ESC **f 6** to turn off ASL. The following message is displayed:

auto screen logging stopped

Screen images are captured when an AID key is pressed.

□ In the synchronous environment, press ENTER, PFn, PAn, CLEAR, ATTN or SYS_REQ.

□ In the asynchronous environment, press ENTER, PF1 to PF8 (corresponding to soft function keys F1 to F8) or CLEAR.

The screen image is logged in the directory defined by the ESCDIR environment variable, in the file named *escort.lg{proc-id}*.

Displaying Screen Attributes

The Displaying Screen Attributes feature shows you attributes of a specific synchronous host application screen, such as protected and unprotected fields. While you are using an application screen in Interactive mode, you can see the attributes on the screen by pressing the special key combination (ESC) **f 5**. Refer to Appendix C for information on translating the characters displayed at the attribute positions on the screen.

The Displaying Screen Attributes feature has no effect in the asynchronous environment.

Local Screen Print

The Local Screen Print feature saves the current screen image. While in Interactive mode in either a host or local session you can save the current screen image by pressing the appropriate key sequence for the PRINT function. Refer to Appendix A for the specific key combination for the PRINT function for your terminal.

The information contained in the screen image is saved in accordance with the following:

Synchronous session Local Screen Print

□ If the 3270 Emulator+ variable *PRNT* is specified, Local Screen Print images are saved in the specified file.

□ If the 3270 Emulator+ variable *PRNT* is not set, Local Screen Print images are saved in the directory defined by the ESCDIR environment variable, in a file named *escort.pr{proc-id}*.

□ Local Screen Print output can be directed using the IDENT function.

Asynchronous session Local Screen Print

□ If the variable *PRNT* is set in your profile, Local Screen Print images are saved in the specified file.

□ If the variable *PRNT* is not set in your profile, Local Screen Print images are saved in the directory defined by the ESCDIR environment variable, in a file named *escort.pr{proc-id}*.

□ Local Screen Print output can be directed using the IDENT function.

Directing Local Screen Print Using IDENT

While in Interactive mode in a host or local session, you can direct Local Screen Print output by pressing the appropriate key sequence for the IDENT function. You can select one of four screen print modes:

□ Press **0** to disable Local Screen Print.

The PRINT function key is disabled and any attempt to use Local Screen Print causes the following message to be displayed in the operator information area:

Screen Print Disabled

□ Press **1** to direct Local Screen Print output to a file.

ESCORT prompts for a pathname; if you do not have write permission for the specified pathname, a message is displayed and you must reenter the required mode number.

□ Press **2** to pipe Local Screen Print output to a UNIX operating system command.

ESCORT prompts for a valid command line; for example,

```
/usr/bin/lp
```

specifies the printer spooler *lp*. Subsequent use of the PRINT function causes Local Screen Print output to be directed to the system default printer.

□ Press **9** to retain the present screen print mode.

Escape to the UNIX Shell

While in Interactive mode in either a host or local session, you can escape to the UNIX shell on the machine on which ESCORT is executing. Press the appropriate key sequence for the SHELL function, to escape from Interactive mode to the UNIX shell. Refer to Appendix A for the specific key combination for the SHELL function for your terminal. Once in the UNIX shell you will be able to perform any of the Restricted Shell mode functions. Press CTRL – **d** or type at the shell prompt

```
exit
```

to return to ESCORT Interactive mode.

This feature is useful, for example, in providing you with the ability to check the contents of a UNIX file to ensure that a download of data from a host system has been completed successfully.

Working in Host and Local Sessions

ESCORT scripts are able to access various host and local sessions. This section describes the concept of multiple sessions and contains the procedures for accessing and displaying these sessions.

Session Identification

ESCORT provides the user with the ability to access a maximum of 10 concurrent sessions from a single script. Sessions are either host or local. Each session in ESCORT has a session identification:

- □ **Host Sessions.** Host sessions may utilize either synchronous or asynchronous communications. ESCORT can access up to eight host applications, of which four may be synchronous and four may be asynchronous.

  ESCORT allows for automatic or interactive switching between sessions and for the exchange of data between applications.

  - The synchronous host session identifications are H1, H2, H3, and H4.
  - The asynchronous host session identifications are A1, A2, A3, and A4.

- □ **Local Sessions.** Up to two local sessions can be connected from an ESCORT script. Local sessions can provide a front-end user interface, for data entry or display.

  - The local session identifications are L1 and L2.

Each session has its own presentation space; presentation space and session are synonymous. Sessions can be considered as virtual terminals, while the presentation spaces are the associated screen buffers for those terminals.

Active and Displayed Sessions in Script Mode

Active Sessions

Only one session can be active at any given time; an active session is any host or local session that is connected to the ESCORT script. All other sessions are dormant. You can change a dormant session into an active session by using the CONNECT command.

The communication ports used in asynchronous host sessions are initialized by using the **SERINIT** command.

All terminal keyboard keys are available for you to use with an active session. However, in relation to all dormant sessions, the keyboard is disabled, with the exception of two special key combinations, QUIT (ESC **f 1**) and SHOW (ESC **f 9**).

These special key combinations are described in more detail later in this section.

When ESCORT is started in Script mode, the synchronous host session H1 is connected and is the active session by default. This session's relevant application screen is also displayed. This default can be changed by specifying a session identification as a parameter in the **PROG** statement. ESCORT commands are discussed in detail in the *ESCORT Programmer's Guide.*

The flexible multiple session features of ESCORT allow you to exchange data easily and quickly between sessions. Data can be manipulated by using reading, writing and searching commands in the active session's presentation space.

A session is made active by use of the **CONNECT** command in a script. The following procedure illustrates how you can move data from one active session to another. In this example, data is moved from local session L1 to synchronous host session H1:

1 Use the **CONNECT** command to make the source session active, for example,

```
CONNECT (L1)
```

2 Capture the desired data.

3 Use the **CONNECT** command to make the target session active, for example,

```
CONNECT (H1)
```

4 Write the previously captured data from the local session to the host session.

Each active or dormant host session in an ESCORT script utilizes a host system connection. Host system connections are a limited resource. You can release host system connections, that are associated with host applications that you no longer need to access in your script, by using the **DISCON** command.

Displayed Sessions

The presentation space of the active session displayed on the terminal can be changed so that any dormant session's presentation space is displayed. You can change the display by using the **SHOW** command.

The **SHOW** and **CONNECT** commands in the script determine which session is displayed and which session is active. Since the displayed and active sessions need not be the same, it is possible to display (**SHOW**) one session while another session is active and possibly executing (through the **CONNECT** command) in the background.

When the displayed and active sessions are not the same, the display on the terminal monitor is not updated, or refreshed, by the active session executing in the background. You can use the **SHOW** command to refresh the terminal monitor display. ESCORT will automatically refresh the terminal monitor display with the active session presentation space when you enter Interactive mode.

Active and Displayed Sessions in Interactive Mode

Active Sessions

ESCORT provides you with the ability to connect or disconnect host sessions in Interactive mode, through the use of six functions. The Interactive mode host session connection functions are

□ connect to next host session, NEXTS

□ connect to previous host session, PREVS

□ open a new synchronous host session, OPENS

□ open a new asynchronous host session, OPENA

□ close a host session, EXIT

□ close all host sessions, QUIT.

NEXTS, PREVS and EXIT are key functions; refer to Appendix A for the specific key combinations for these functions for your terminal.

OPENS, OPENA and QUIT are ESCORT key functions; refer to "Special Key Combinations" at the end of this chapter for the specific key combinations for these functions.

The following describes the operation of each host connection function in Interactive mode.

The cycling order of host sessions is synchronous connections, followed by asynchronous connections. In the hierarchy of these connections, H1 is the lowest and H4 the highest synchronous host sessions, and A1 the lowest and A4 the highest asynchronous host sessions.

NEXTS **Connects to next host session.**

If you press the NEXTS key combination, ESCORT connects to and makes active the previously opened, next higher host session.

For example, your ESCORT script has opened host sessions H1, H3, A1, and A2. Synchronous host session H1 is the active (connected) and displayed session when you enter Interactive mode, sessions H3, A1 and A2 remain dormant. If you now press the NEXTS key combination, synchronous host session H3 is connected and displayed. Repeated use of the NEXTS key combination, in this example, connects and displays the asynchronous sessions A1, A2, and then the synchronous session H1, and so on.

If only one session is connected when you enter Interactive mode, use of the NEXTS key combination displays the following message in the operator information area:

No other host session active

PREVS **Connects to previous host session.**

The PREVS key combination works in a similar manner to the NEXTS function, however, host sessions are connected and displayed in descending order.

Using the example detailed in NEXTS, pressing the PREVS key combination in Interactive mode when connected to synchronous host session H1 connects and displays the asynchronous session A2.

Similarly, if only one session is connected when you enter Interactive mode, pressing the PREVS key combination displays the following message in the operator information area:

No other host session active

OPENS **Opens new synchronous host session.**

Pressing the OPENS key combination opens the next available higher synchronous host session.

For example, if synchronous host session H1 is the active session when you enter Interactive mode, synchronous host session H3 being dormant, and you press the OPENS key combination, ESCORT opens synchronous host session H2. This session then becomes the active session. If the OPENS key combination is pressed again, in this example, ESCORT opens the next available higher synchronous host session, H4.

ESCORT displays the following message in the operator information area when you press the OPENS key combination:

```
Connecting to new synchronous session
```

If all four synchronous host sessions are connected, and you attempt to open a further synchronous host session by pressing the OPENS key combination, the following message is displayed:

```
Max. number of synchronous sessions already
active
```

If no host system logical unit connections are available when you attempt to open a synchronous host session using OPENS, or the logical unit open failed, the following message is displayed:

```
Unable to open new synchronous session
```

OPENA **Opens new asynchronous host session.**

Pressing the OPENA key combination opens the next available higher asynchronous host session. You are prompted for the asynchronous communication port initialization parameters, the defaults for which are the last parameters established either interactively or via a **SERINIT** command in a script.

To abort the asynchronous host connection after the communication port initialization parameters have been entered but before the connection is established, press the EXIT key combination.

For example, if asynchronous host session A1 is the active session when you enter Interactive mode and asynchronous host session A4 (being the last session established) is dormant, and you press the OPENA key combination, ESCORT opens asynchronous host session A2. You are prompted for the asynchronous communication port initialization parameters (speed, parity, stopbits, length, duplex, telephone number/machine name, TTY port, and flow control). By default, the initialization parameters for asynchronous host session A4 are displayed; these defaults can be amended. Asynchronous host session A2 then becomes the active session.

Repeated use of the OPENA key combination, in this example, opens and makes active the asynchronous session A3.

ESCORT displays the following message in the operator information area when you press the OPENA key combination:

Connecting to new asynchronous session

If all four asynchronous host sessions are connected, and you attempt to open a further asynchronous host session by pressing the OPENA key combination, the following message is displayed:

Max. number of asynchronous sessions already active

If ESCORT is unable to establish a connection when you attempt to open an asynchronous host session using OPENA, the following message is displayed:

Unable to open new asynchronous session

ESCORT writes specific error messages to the *escort.pr{proc-id}* file, created in the directory defined by the ESCDIR environment variable.

Some asynchronous applications request terminal type information. You should specify your terminal as a VT100 on these remote asynchronous hosts, regardless of the actual terminal type being used.

EXIT **Closes a host session.**

The EXIT key combination closes the active host session and automatically activates the lowest host session, if any, within the same environment. If no other host session is available in the same environment, the lowest available host session is activated.

If synchronous host sessions H2 and H3, and asynchronous host sessions A1 and A2 are open, for example, and you press the EXIT key combination in Interactive mode when synchronous host session H3 is active, this session is terminated, the following message is displayed briefly:

```
TERMINAL SESSION TERMINATED
```

and synchronous host session H2 becomes the active session.

Repeated use of the EXIT key combination, in this example, closes the synchronous host session H2 and makes active the asynchronous host session A1, and so on.

If the EXIT function is used when only one host session is connected, the ESCORT process is terminated.

QUIT **Close all host sessions.**

The QUIT key combination terminates all connected host sessions, quits ESCORT and returns to the UNIX shell. The following message is displayed briefly:

```
TERMINAL SESSION TERMINATED
```

Caution

You should ensure that the status of host sessions (that is, which sessions are open but dormant and which session is active) on resuming a script from Interactive mode, duplicates the status at the point when script execution was interrupted by entering Interactive mode. Script execution may be affected and results may be unpredictable if the script anticipates host sessions to be available and the status of the sessions has been altered in any way.

Displayed Sessions

While in Interactive mode in a host session, or Tutorial mode in a host or local session, you can display another session's presentation space by using the SHOW special key combination. To SHOW the next presentation space press ESC **f 9**.

Each time the SHOW key combination is pressed the next presentation space is displayed in a circular manner. For example, if the synchronous host session H1 presentation space is currently displayed and you press the SHOW key combination (ESC **f 9**) the synchronous host session H2 presentation space is displayed.

Repeated use of the SHOW key combination displays other synchronous and asynchronous host, and local sessions, if any, and eventually redisplays the presentation space for the H1 synchronous host session.

The SHOW function only displays the appropriate session and does not change the active session. The session identification of the displayed session is shown in the operator information area and the session identification is highlighted if the displayed session and the active session are the same. See "Status Line Information" in this section for more information.

Status Line Information

The session identification of the session that you display when you use the **SHOW** command in a script or the SHOW key combination (ESC **f 9**), is displayed in the operator information area on the terminal. The session identification is highlighted if the displayed session and the active session are the same.

If your terminal has a 24-line screen, the operator information area, or status line, must first be toggled on using the appropriate key sequence for the STAT function. Refer to Appendix A for the specific key combination for the STAT function for your terminal. The operator information area is automatically displayed on terminals with 25-line screens.

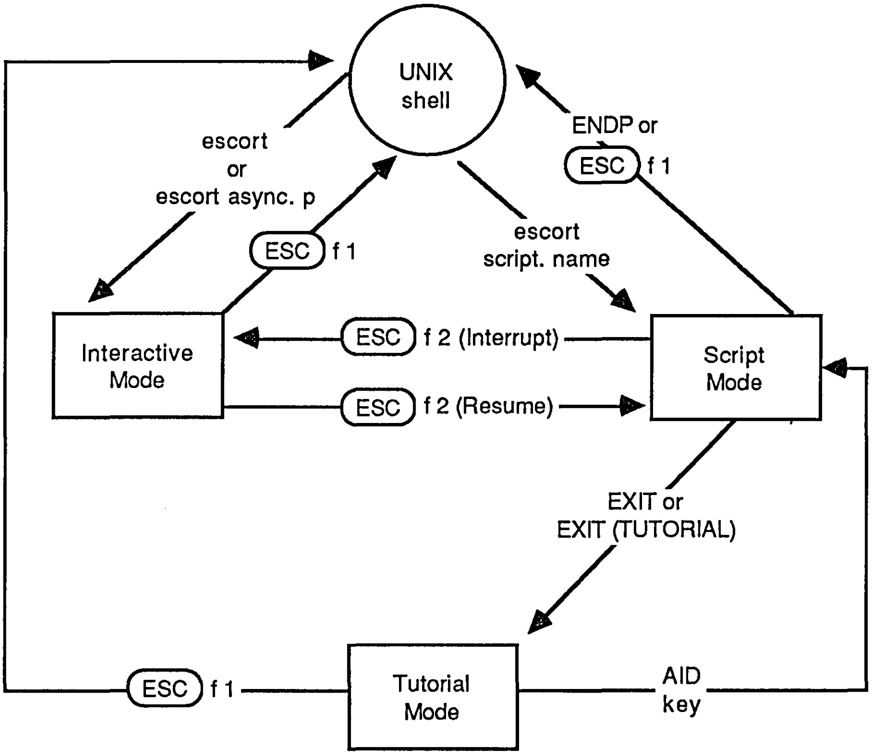
The mode of operation is also displayed in the operator information area on the terminal by use of the following codes: *I* for Interactive mode, *S* for Script mode and *T* for Tutorial mode.

Modes of Operation Summary

The following diagrams illustrate the relationship between Script, Interactive and Tutorial modes, and summarize the Interactive mode special features and the methods of changing active sessions and displays.

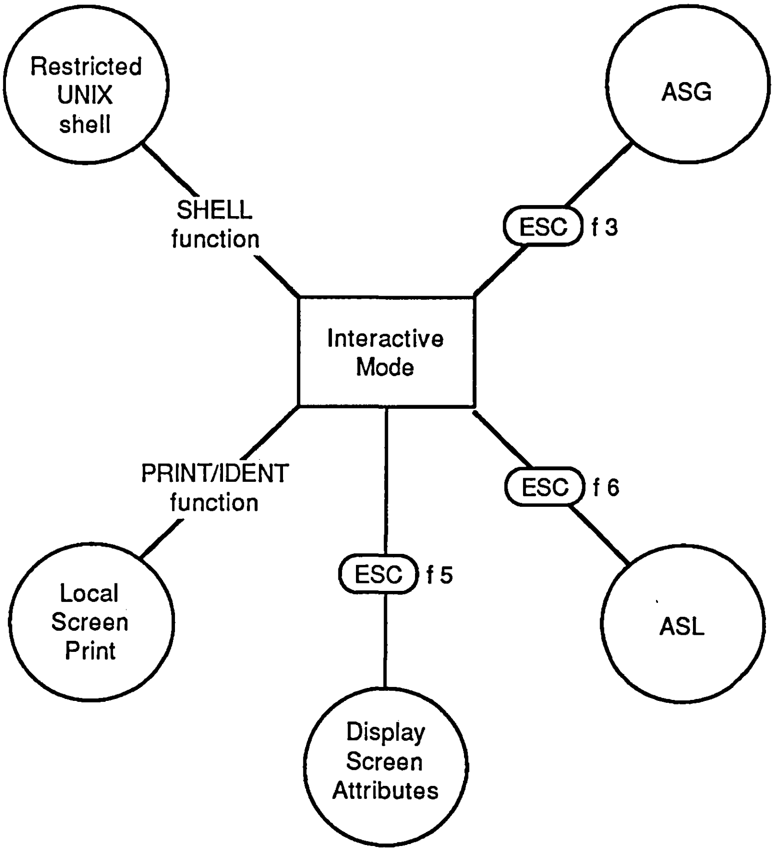
Changing Modes

FIGURE 2-5



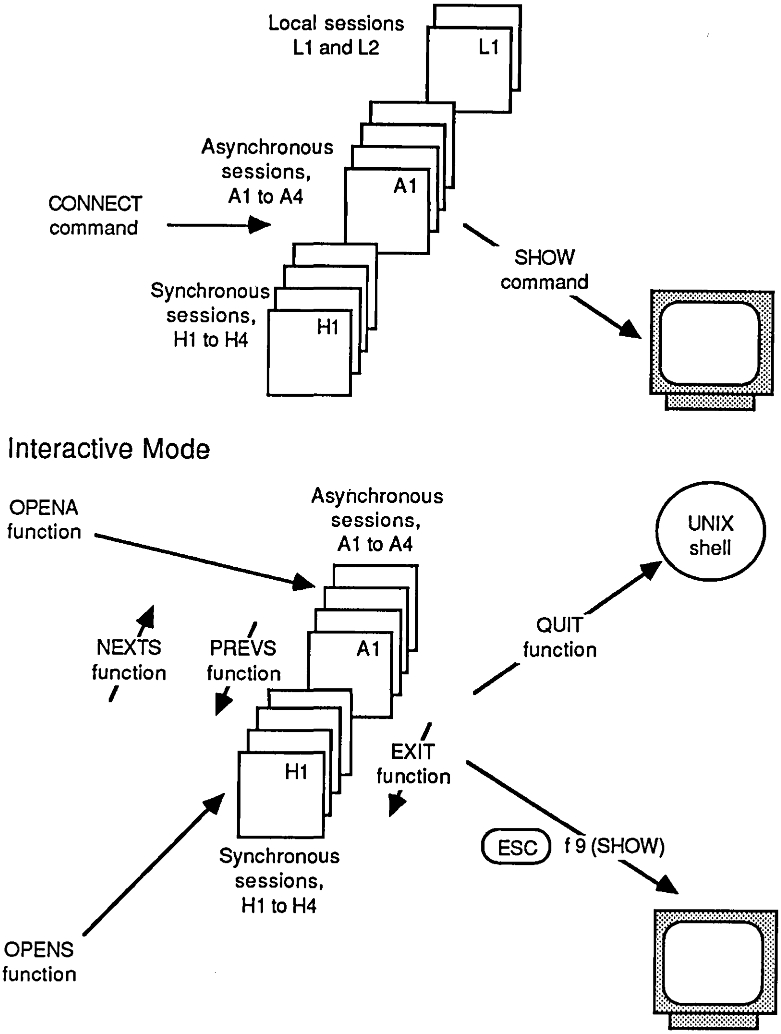
Interactive Mode Special Features

FIGURE 2-6



Active and Displayed Sessions

FIGURE 2-7



Script Mode

Interactive Mode

ESCORT Utility Files

This section describes five utility files that ESCORT automatically creates in the directory defined by the ESCDIR environment variable. Refer to the section, "Directory Environment Variable" in Appendix B for information on setting the directory environment variable.

The five utility file names have the following format

escort.{*type*}{*proc-id*}

The five {*type*}s produced are

cp Data saved as a result of a **CAPTURE ON** command in a script is written to this file type.

dp A **DUMP** command in a script or a fatal error produces an ESCORT dump to this file type.

ky In Automatic Script Generation (ASG) mode the generated script is automatically written to this file type. See "Automatic Script Generation" in this section for more information.

lg In Automatic Screen Logging (ASL) mode screen images are automatically written to this file type. See "Automatic Screen Logging" in this section for more information.

Data specified by a **LOG** command in a script is written to this file type.

pr Data specified by a **PRINT** command in a script is written to this file type.

If an error occurs during syntax checking or script execution, ESCORT will write certain error message information to a *pr* type utility file.

The identifier, {*proc-id*}, refers to the unique process identification number that the UNIX operating system automatically assigns to each process.

For example the file named

escort.ky23658

contains a script automatically generated by ESCORT through the Automatic Script Generation feature; the UNIX operating system assigned the unique process identification number *23658* to the process.

When an ESCORT session is terminated, the process identification number is displayed on the terminal, for example

```
Process Id of your ESCORT session was: 23658
```

Use the process identification number to access the various utility files created by ESCORT during the session.

Command Line

The ESCORT command line allows you to

□ operate interactively

□ run a script

□ perform syntax checking on a script without execution, and create an executable run-time script.

The syntax of the command line is

```
escort [script_name [p1,p2,....,p9] [NORUN] [NOLOGO]]
```

where the items enclosed in brackets are optional.

The ***script_name*** is the name of a script file. You can specify the full path for the script; for instance,

```
/usr/escort/slib/upload
```

The ***p1,p2....,p9*** are parameters passed to the script from the command line. You may pass up to nine parameters on the command line. You must not insert any blanks between parameters. Parameters may be passed when you run (parse and interpret) a non-run-time script. See the section "Parameter Passing" in Chapter 2 in the *ESCORT Programmer's Guide* for further information.

The ***NORUN*** keyword parameter is used to create an executable run-time version of a script. You may specify the parameter in lowercase letters. When you execute a script with the ***NORUN*** parameter, the script is parsed only (that is, the syntax is checked) and, provided no syntax errors occur, a new script is created with the file extension, *x*, in the form *script_name.x*.

This new script is a binary file and the parsing phase is eliminated when this script is executed. Eliminating parsing, and interpreting the script immediately, saves time.

No parameters may be passed to the script from the command line when executing a run-time script. Use one of the following options if you need to pass parameters when executing a run-time script:

- □ Set UNIX environment variables to the value of the parameters and read these environment variables from within your script using the **$GETENV** function.
- □ Store the required parameters in a file and read the contents of the file using the **READ** command.

The ***NOLOGO*** keyword parameter is used to eliminate the banner screen. This keyword may be specified in lowercase letters.

Examples of using the command line follow:

escort	Enter Interactive mode
escort script_name	Parse and interpret (execute) a script
escort script_name NORUN	Parse script and create run-time script
escort script_name.x	Interpret (execute) run-time script
escort script_name p1,p2,p3	Parse and interpret a script and pass 3 parameters
escort script_name p1,p2,p3 NORUN	Parse script, pass 3 parameters and create run-time script
escort script_name p1,p2,p3 NOLOGO	Parse and interpret script, pass 3 parameters and eliminate banner
escort script_name 2 > /usr/errors	Parse and interpret script, redirect standard error to specified file
escort script_name 1 > /dev/null &	Parse and interpret script in background, and redirect standard output

Special Key Combinations

Use of the default special key combinations, ESC (Escape), and letter **f** and numeric keys, or ESC and letters **o**, **a** and **s** keys, simulate function keys with special meaning in ESCORT.

To access these functions press the following sequentially:

- □ ESC, the letter **f** key and the appropriate numeric key,
- □ ESC, the letter **o** key and the letter **s** key,
- □ ESC, the letter **o** key and the letter **a** key.

The list of special key combinations is also contained in your Quick Reference Card.

The use of the special key combinations is summarized below. Most of the key combinations have been explained in greater detail earlier in this chapter.

If the ESCORT default key combinations have been amended by your System Administrator, you must substitute the amended key combinations for the default combinations shown in this section.

ESC **f 1**

Quit (QUIT)

Terminates ESCORT execution and returns to the UNIX shell with a default exit code 0. All open files are closed. For example, the files *escort.lg*{*proc-id*} and *escort.ky*{*proc-id*} are automatically closed. ESC **f 1** may be used from either Interactive mode when connected to a host session, or Script mode when connected to a host or local session.

ESC **f 2**

Interrupt/Resume (I/R)

Interrupts active host script execution and enters Interactive mode. You may also use this key combination to resume script execution.

ESC **f 3**

Automatic Script Generation (ASG)

Activates or deactivates automatic script generation (ASG). ESC **f 3** may be used only from Interactive mode in an active host session. After you press ESC **f 3** ESCORT generates a script containing all keyboard entries after the ESC **f 3** was pressed. You may deactivate automatic script generation by pressing ESC **f 3**. The automatically generated script appears in the *escort.ky{proc-id}* file.

ESC **f 4**

Cursor Position (CURSR_POS)

Displays the current cursor position in the operator information area. Both the absolute position and row and column position appear. ESC **f 4** may be used from Interactive mode in a synchronous host session or from Tutorial mode in a local session.

ESC **f 4** has no effect when the active session is an asynchronous session.

ESC **f 5**

Display Attributes (ATTRIB)

Displays all field attributes on the screen. ESC **f 5** may be used from Interactive mode in a synchronous host session or from Tutorial mode in a local session.

ESC **f 5** has no effect when the active session is an asynchronous session.

ESC **f 6** **Automatic Screen Logging (ASL)**

Activates or deactivates automatic screen logging. The *escort.lg{proc-id}* file contains the screen images. ESC **f 6** may be used from Interactive mode in a host session or from Tutorial mode in a local session.

ESC **f 7** **Key Status (KEY_STATUS).**

Displays the status of the special keys for Automatic Script Generation, Automatic Screen Logging, and AID Substitution. ESC **f 7** may be used only in Interactive mode in an active host session.

ESC **f 8** **No effect.**

This key combination has no effect in ESCORT.

ESC **f 9** **Show Session (SHOW).**

Displays the next session on the terminal in circular manner. Use of this function does not change the active session. ESC **f 9** may be used in either Interactive mode in an active host session or Tutorial mode in an active host or local session.

ESC **f 0** **AID Substitution (AID_SUB).**

Activates or deactivates AID subroutine substitution while in Automatic Script Generation. ESC **f 0** may be used only while you are in Automatic Script Generation in Interactive mode in an active synchronous host session. Each time an AID key is encountered in the automatically generated script, ESCORT generates a subroutine call to a script named *aid_resp*.

This feature is required for synchronous host applications that have no-response mode transactions.

For further information on AID key subroutines, see the section "Synchronous Response/No-Response Mode Transactions" in Chapter 2, and the appendix, "AID Subroutines Library" in the *ESCORT Programmer's Guide.*

ESC **f 0** has no effect when the active session is an asynchronous session.

ESC **o s**

Open Synchronous Host Session (OPENS).

Opens the next available higher synchronous host session. ESC **o s** may be used in Interactive mode while connected to either a synchronous host or an asynchronous host session. The synchronous host session opened when you press ESC **o s** becomes the connected session.

ESC **o a**

Open Asynchronous Host Session (OPENA).

Opens the next available higher asynchronous host session. ESC **o a** may be used in Interactive mode while connected to either a synchronous host or an asynchronous host session. The asynchronous host session opened when you press ESC **o a** becomes the connected session.

Appendices

Appendices

Key Sequences

This appendix lists the key sequences that emulate IBM 3278 key functions for the following terminals:

Synchronous Terminals

□ Standard ASCII terminals

□ AT&T 4410 and Teletype® 5410 terminals

□ AT&T 4418 and Teletype 5418 terminals

□ AT&T 4425 and Teletype 5425 terminals

□ AT&T 605 Business Communications Terminal (BCT)

□ AT&T 610, 615, 620, and 630 Business Communications Terminals (BCTs).

Asynchronous Terminals

□ DEC VT100 terminal.

Note that certain key functions are ignored if either ESCORT or the terminal does not support them. The following key functions are *not* supported by ESCORT:

ALLCAP
ALT_CR
BLINK
BOT
CAN
CLICK
COLR
CTRL
CURSR_SEL
NULLEND
NUM_OV
TOP

The synchronous keyboard files are the same as the keyboard files used by the AT&T 3270 Emulator+ software.

Key Sequences for Standard ASCII Terminals

The AT&T 4415, 5420 and the Tektronix™ 4105 terminals use these key sequences.

3278 Key Function	Standard ASCII Terminal Key Sequence
ATTN	ESC **a** RETURN
BAKTAB	CTRL **e**
BS	CTRL **h**
CENT	**^**
CLEAR	ESC **z**
DEL	DEL
DEV_CNCL	ESC **d**
DOWN_A	CTRL **v**
DUP	CTRL **d**
E_EOF	ESC **e f**
E_INPUT	ESC **e i**
ENTER	RETURN
ENTER1	RETURN
EXIT	ESC **x x**
FM	CTRL **k**
HOME	CTRL **o**
IDENT	ESC **i**
INS	CTRL **u**
LDUB	CTRL **r**
LEFT_A	CTRL **f**
NEWL	CTRL **j**
NEXTS	ESC **+**
NOT	**]**
PA1 to PA3	ESC **a** *key number* RETURN
PF1 to PF24	ESC *key number* RETURN
PREVS	ESC **–**
PRINT	ESC **p**
RDUB	CTRL **y**
REDRAW	ESC **r**
RESET	CTRL **a**
RIGHT_A	CTRL **g**

Key Sequences for Standard ASCII Terminals (continued)

3278 Key Function	Standard ASCII Terminal Key Sequence
SHELL	ESC **s**
SOLID	**[**
STAT	ESC **l**
SYS_REQ (SNA only)	ESC **q**
TAB	CTRL **i**
TEST_REQ (BSC only)	ESC **q**
UP_A	CTRL **t**

Key Sequences for AT&T 4410 and Teletype 5410 Terminals

3278 Key Function	AT&T 4410 and Teletype 5410 Key Sequence
ATTN	ESC **a** RETURN
BAKTAB	↙
BS	CTRL **^ h**
CENT	**^**
CLEAR	ESC **z**
DEL	DEL
DEV_CNCL	ESC **d**
DOWN_A	↓
DUP	CTRL **d**
E_EOF	ESC **e f**
E_INPUT	ESC **e i**
ENTER	RETURN
ENTER1	RETURN
EXIT	ESC **x x**
FM	CTRL **k**
HOME	↖
IDENT	ESC **i**
INS	CTRL **u**
LDUB	CTRL **r**
LEFT_A	←
NEWL	CTRL **j**
NEXTS	ESC **>**
NOT	**]**
PA1 to PA3	ESC **a** *key number* RETURN
PF1 to PF9	ESC *key number*
PF10	ESC **0**
PF11	ESC **–**
PF12	ESC **=**
PF13	ESC SHIFT **1**
PF14	ESC SHIFT **2**
PF15	ESC SHIFT **3**

Key Sequences for AT&T 4410 and Teletype 5410 Terminals (continued)

3278 Key Function	AT&T 4410 and Teletype 5410 Key Sequence
PF16	ESC SHIFT 4
PF17	ESC SHIFT 5
PF18	ESC SHIFT 6
PF19	ESC SHIFT 7
PF20	ESC SHIFT 8
PF21	ESC SHIFT 9
PF22	ESC SHIFT 0
PF23	ESC SHIFT -
PF24	ESC SHIFT =
PREVS	ESC <
PRINT	ESC p
RDUB	CTRL y
REDRAW	ESC r
RESET	CTRL a
RIGHT_A	→
SHELL	ESC s
SOLID	[
STAT	ESC l
SYS_REQ (SNA only)	ESC q
TAB	CTRL i
TEST_REQ (BSC only)	ESC q
UP_A	↑

Key Sequences for AT&T 4418 and Teletype 5418 Terminals

Note that on these terminals there is no key marked ESC or CTRL.

To emulate ESC, press ALT and |.

For CTRL use the key immediately to the left of the space bar.

3278 Key Function	AT&T 4418 and Teletype 5418 Key Sequence
ATTN	ATTN
BAKTAB	BACK TAB
BS	BACK SPACE
CENT	^
CLEAR	CLEAR
DEL	DEL
DEV_CNCL	DEV CNCL
DOWN_A	↓
DUP	DUP
E_EOF	ERASE EOF
E_INPUT	ERASE INPUT
ENTER	ENTER (lower right side)
ENTER1	ENTER (upper left side)
EXIT	ESC **x x**
FM	FIELD MARK
HOME	HOME
IDENT	IDENT
INS	INS
LDUB	← ←
LEFT_A	←
NEWL	NEW LINE
NEXTS	ESC +
NOT	]

Key Sequences for AT&T 4418 and Teletype 5418 Terminals (continued)

3278 Key Function	AT&T 4418 and Teletype 5418 Key Sequence
PA1	PA1
PA2	PA2
PA3	SHIFT INS
PF1 to PF24	PF1 to PF24
PREVS	ESC –
PRINT	PRINT LCL
RDUB	→ →
REDRAW	ESC **r**
RESET	RESET (lower left side)
RIGHT_A	→
SHELL	SHIFT RESET (upper left side)
SOLID	**[**
STAT	RESET (upper left side)
SYS_REQ (SNA only)	SYS REQ
TAB	CURSOR TAB
TEST_REQ (BSC only)	SYS REQ
UP_A	↑

Key Sequences for AT&T 4425 and Teletype 5425 Terminals

3278 Key Function	AT&T 4425 and Teletype 5425 Key Sequence
ATTN	ESC **a** RETURN
BAKTAB	SHIFT TAB
BS	BACK SPACE
CENT	^
CLEAR	CLEAR
DEL	DEL
DEV_CNCL	ESC **d**
DOWN_A	↓
DUP	CTRL **d**
E_EOF	CLEAR LINE
E_INPUT	DELETE LINE
ENTER	RETURN
ENTER1	ENTER (on keypad)
EXIT	ESC **x x**
FM	CTRL **k**
HOME	HOME
IDENT	ESC **i**
INS	INSERT CHAR
LDUB	CTRL **r**
LEFT_A	←
NEWL	CTRL **j**
NEXTS	ESC +
NOT	]
PA1 to PA3	ESC **a** *key number* RETURN
PF1 to PF4	PF1 to PF4
PF5	**7** (on keypad)
PF6	**8** (on keypad)
PF7	**9** (on keypad)
PF8	– (on keypad)
PF9	**4** (on keypad)
PF10	**5** (on keypad)

Key Sequences for AT&T 4425 and Teletype 5425 Terminals (continued)

3278 Key Function	AT&T 4425 and Teletype 5425 Key Sequence
PF11	**6** (on keypad)
PF12	**,** (on keypad)
PF13 to PF24	ESC *key number* RETURN
PREVS	ESC **–**
PRINT	ESC **p**
RDUB	CTRL **y**
REDRAW	ESC **r**
RESET	CTRL **a**
RIGHT_A	→
SHELL	ESC **s**
SOLID	**[**
STAT	ESC **l**
SYS_REQ (SNA only)	ESC **q**
TAB	TAB
TEST_REQ (BSC only)	ESC **q**
UP_A	↑

Key Sequences for AT&T 605 Business Communications Terminal

The AT&T 605 Business Communications Terminal (BCT) has a 102-key keyboard.

3278 Key Function	AT&T 605 BCT Key Sequence
ATTN	ESC **a** RETURN
BAKTAB	SHIFT TAB
BS	BACK SPACE
CENT	**^**
CLEAR	SHIFT CLEAR
DEL	CTRL DELETE
DEV_CNCL	ESC **d**
DOWN_A	↓
DUP	CTRL **d**
E_EOF	ESC **e f**
E_INPUT	SHIFT DEL LN
ENTER	RETURN
ENTER1	RETURN
EXIT	ESC **x x**
FM	CTRL **k**
HOME	CLEAR HOME
IDENT	ESC **i**
INS	INS LN
LDUB	CTRL **r**
LEFT_A	←
NEWL	CTRL **j**
NEXTS	ESC **+**
NOT	**]**
PA1 to PA3	ESC **a** *key number* RETURN
PF1 to PF24	ESC *key number* RETURN
PREVS	ESC **–**
PRINT	ESC **p**
RDUB	CTRL **y**
REDRAW	ESC **r**
RESET	ESC **c**
RIGHT_A	→

Key Sequences for AT&T 605 Business Communications Terminal (continued)

3278 Key Function	AT&T 605 BCT Key Sequence
SHELL	ESC **s**
SOLID	[
STAT	ESC **l**
SYS_REQ (SNA only)	ESC **q**
TAB	TAB
TEST_REQ (BSC only)	ESC **q**
UP_A	↑

Key Sequences for AT&T 610, 615, 620, and 630 Business Communications Terminals

The AT&T 610, 615, 620, and 630 Business Communications Terminals (BCTs) have 98-key keyboards.

3278 Key Function	AT&T 610, 615, 620, and 630 BCTs Key Sequence
ATTN	ESC **a** RETURN
BAKTAB	SHIFT TAB
BS	BACK SPACE
CENT	^
CLEAR	CLEAR
DEL	DELETE
DEV_CNCL	ESC **d**
DOWN_A	↓
DUP	CTRL **d**
E_EOF	ESC **e f**
E_INPUT	ESC **e i**
ENTER	RETURN
ENTER1	RETURN
EXIT	ESC **x x**
FM	CTRL **k**
HOME	HOME
IDENT	ESC **i**
INS	CTRL **u**
LDUB	CTRL **r**
LEFT_A	←
NEWL	CTRL **j**
NEXTS	ESC +
NOT	]
PA1 to PA3	ESC **a** *key number* RETURN
PF1 to PF24	ESC *key number* RETURN
PREVS	ESC –
PRINT	ESC **p**
RDUB	CTRL **y**
REDRAW	ESC **r**
RESET	ESC **c**
RIGHT_A	→

Key Sequences for AT&T 610, 615, 620, and 630 Business Communications Terminals (continued)

3278 Key Function	AT&T 610, 615, 620, and 630 BCTs Key Sequence
SHELL	ESC s
SOLID	[
STAT	ESC l
SYS_REQ (SNA only)	ESC q
TAB	TAB
TEST_REQ (BSC only)	ESC q
UP_A	↑

Key Sequences for DEC VT100 Terminal

DEC VT100 Key Function	ESCORT Equivalent Key Sequence
BAKTAB	CTRL **e**
BS	CTRL **h**
CLEAR	ESC **z**
DEL	DEL
DOWN_A	CTRL **v**
ENTER	RETURN
ENTER1	RETURN
EXIT	ESC **x x**
HOME	CTRL **o**
IDENT	ESC **i**
LEFT_A	CTRL **f**
NEWL	CTRL **j**
NEXTS	ESC **+**
PF1 to PF8	ESC *key number* RETURN
PREVS	ESC **–**
PRINT	ESC **p**
REDRAW	ESC **r**
RIGHT_A	CTRL **g**
SHELL	ESC **s**
STAT	ESC **l**
TAB	CTRL **i**
UP_A	CTRL **t**

Note

This key sequence table should be used in asynchronous-only environments. If you communicate with both synchronous and asynchronous hosts, use the appropriate key sequence table for your specific synchronous terminal for all synchronous and asynchronous applications.

Environment Variables

This appendix provides information on setting environment variables applicable to your operating environment.

Setting 3270 Emulator+ ESCORT Environment Variables

Once the ESCORT software has been installed, certain prerequisite variables should be set in your *profile* file. When invoking ESCORT, ensure that the 3270 Emulator+ terminal manager process is running.

Terminal Environment Variable

ESCORT uses the environment variable, *TERM*, to access terminal information in the system file *terminfo* for screen management. The following example shows the environment variable set for an AT&T 4410 terminal type.

```
TERM=4410
export TERM
```

3270 Emulator+ Environment Variables

ESCORT runs in conjunction with the AT&T 3270 Emulator+ software. Your *profile* file should be edited to include the following command:

```
. /usr/bscadm/runtime/bscenvset
```

or

```
. /usr/snaadm/runtime/snaenvset
```

Set the appropriate environment variable so that 3270 Emulator+ and ESCORT will execute properly.

D3274 Environment Variable

The default value for the D3274 environment variable provided by the *snaenvset* command allows access to all available logical unit connections. Setting the D3274 environment variable provides controlled access to certain host applications. You can assign ranges of logical unit ports to particular users. In the following example, a user is given access to eight logical unit ports.

```
D3274=1-5,13,14,15
export D3274
```

Host/Local Session Environment Variable

The UNIX operating system environment variable, *ESCHOST*, determines whether a synchronous connection is to be established. The environment variable can be set to 0 or 1; if set to a value of 1 (the default value if this variable is not set) the ESCORT script will be able to connect to a synchronous host session.

Setting the ESCHOST environment variable to 0 is useful

- in limiting access to prototyping local screen formats
- if ISC or SDLI cards have not been installed in the 3B processor
- if the user accesses only asynchronous host applications.

In the following example, a user's ESCORT connections will default to a local session.

```
ESCHOST=0
export ESCHOST
```

Directory Environment Variable

The UNIX operating system environment variable, *ESCDIR*, determines the path for the five types of ESCORT utility files. If the *ESCDIR* environment variable is not set, ESCORT utility files are created in your *$HOME* directory. In the following example, a user's ESCORT utility files will be created in a directory named *sys_1*, a subdirectory of */usr/john*.

```
ESCDIR=/usr/john/sys_1
export ESCDIR
```

Terminal Information Environment Variable

To use the *terminfo* terminal information files installed by ESCORT, set the *terminfo* variable as follows:

```
TERMINFO=/usr/escort/terminfo
export TERMINFO
```

Setting this environment variable is only necessary if the system-supplied files contain errors or have been modified in some way and ESCORT does not function correctly.

Interpretation of Attribute Bytes

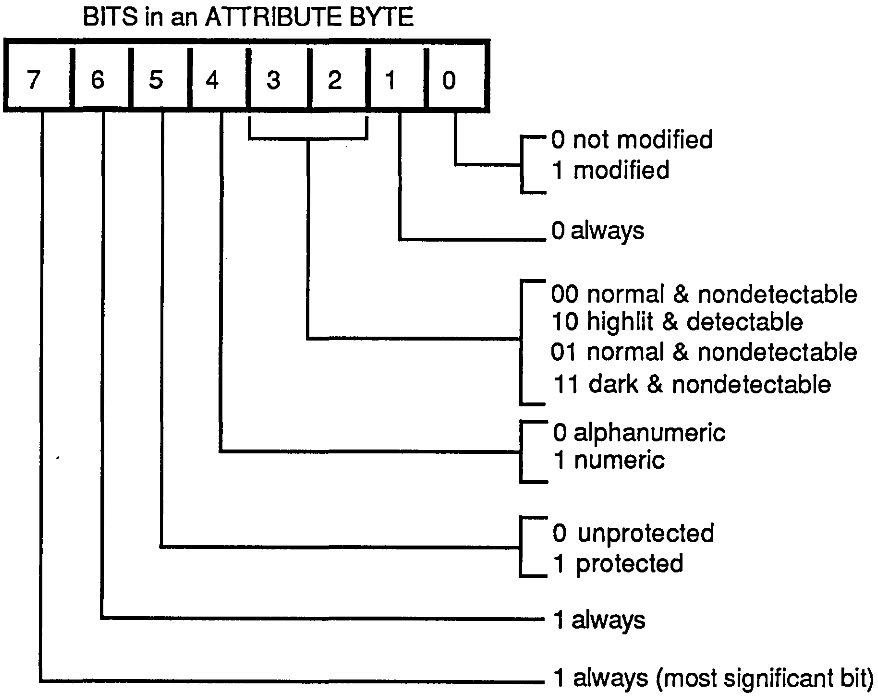
When you press ESC **f 5** while connected to a synchronous host session, ESCORT displays an alpha character in the position of each attribute byte on the screen. The character displayed represents the Primary Attributes for each field; this appendix shows you how to interpret the displayed characters.

The following table shows you how to convert the character displayed on the terminal screen into attribute bytes.

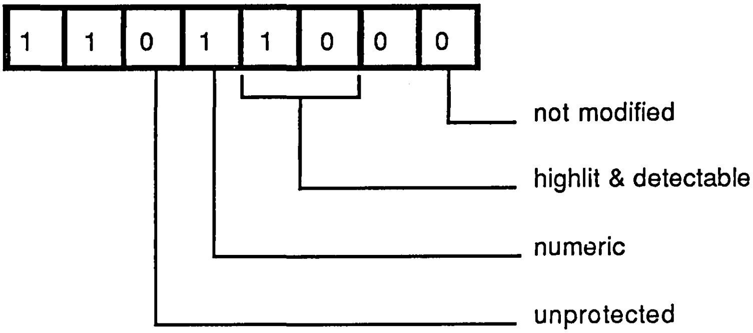
ASCII Character to Attribute Byte Conversion

Char	Attr	Char	Attr	Char	Attr	Char	Attr
@	1100 0000	P	1101 0000	'	1110 0000	p	1111 0000
A	1100 0001	Q	1101 0001	a	1110 0001	q	1111 0001
B	1100 0010	R	1101 0010	b	1110 0010	r	1111 0010
C	1100 0011	S	1101 0011	c	1110 0011	s	1111 0011
D	1100 0100	T	1101 0100	d	1110 0100	t	1111 0100
E	1100 0101	U	1101 0101	e	1110 0101	u	1111 0101
F	1100 0110	V	1101 0110	f	1110 0110	v	1111 0110
G	1100 0111	W	1101 0111	g	1110 0111	w	1111 0111
H	1100 1000	X	1101 1000	h	1110 1000	x	1111 1000
I	1100 1001	Y	1101 1001	i	1110 1001	y	1111 1001
J	1100 1010	Z	1101 1010	j	1110 1010	z	1111 1010
K	1100 1011	[	1101 1011	k	1110 1011	{	1111 1011
L	1100 1100	\	1101 1100	l	1110 1100	|	1111 1100
M	1100 1101	]	1101 1101	m	1110 1101	}	1111 1101
N	1100 1110	^	1101 1110	n	1110 1110	—	1111 1110
O	1100 1111	—	1101 1111	o	1110 1111	<	1111 1111

The following diagram shows you how to read the bits in an attribute byte. Note that bits 3 and 2 are coupled and are read together. *Detectable* refers to detectable by a light pen.



For example, if the alphabetic character X is displayed when the Display Attribute key sequence (ESC **f 5**) is pressed, the corresponding attribute byte is 1101 1000. Interpretation of the attribute byte shows that the field has the following characteristics:



Glossary

Glossary

This glossary contains definitions for terms and acronyms used throughout this guide. These terms are defined according to their meaning in ESCORT and may not have the same meaning in other programming languages.

Active session Any host or local session connected to a script or connected interactively, to which all ESCORT commands are directed.

Administrative command A command that shows where a program, subroutine, or comment begins or ends. **PROG** is an administrative command.

AID key Attention-identifier key. AID key commands simulate the action of one of the attention-identifier keys. ENTER is an example of an AID key.

Arithmetic operator A character (such as +) that represents a mathematical operation.

Array A collection of values of the same type referred to by a single name. Each entry in an array is called an element.

Automatic screen logging A feature of ESCORT that saves the image of a specific application screen and any data entered on it. Also called ASL.

Automatic script generation A feature of ESCORT that automatically creates a script from a user's terminal session. Also called ASG.

Concatenation The operation that joins two strings together.

Constant A fixed value or data item. A constant may be a string or a numeric constant.

Debugging command A programming aid used to check for errors or to detect failures in program execution. **DUMP** is a debugging command.

Declaration A statement that defines the type and amount of data associated with a symbolic label. Declaration commands include **INT** and **CHAR.**

Default The value or option that is assumed when none is given. For example, if you use the command **BTAB** and do not specify the number of back-tabs to be executed, ESCORT assumes the value is 1.

Destination The variable to the left of the equal sign in an assignment statement.

Emulator The ESCORT component that allows an ASCII terminal to perform the functions of an IBM 3278 terminal in the synchronous environment, or a DEC VT100 terminal in the asynchronous environment.

Expression A single operand or multiple operands separated by operators.

Extended Field Attributes Three arguments used in a **FIELD** statement that define advanced screen characteristics similar to those found in IBM synchronous host screen formats. Also called EFA.

Field variable An area of the screen buffer defined by the user and assigned a symbolic name. Also called screen field variable.

File management command A command used to open or close a file or to control the input and output of data from a file. **READ** is a file management command.

Format A symbolic name for a group of fields that constitute a screen.

Function An algorithm that returns an integer or string value. Function names in ESCORT are preceded by a $.

Global variable A variable that is accessible throughout a program.

Host session The connection of your ASCII terminal through the 3B processor to a synchronous or an asynchronous host computer, providing you with access to a host computer application.

ISC card Intelligent Serial Controller card. A card installed in the 3B2 processor that provides communications ability to synchronous host computers.

Interactive mode An ESCORT feature that allows you to use your ASCII terminal as a synchronous IBM 3278 terminal, or an asynchronous DEC VT100 terminal.

Interpreter The ESCORT component that executes a script.

Keyword An operand predefined by ESCORT, such as *SCREEN*.

Label A word or symbol used at the beginning of a program statement to branch from a **GOTO** statement.

Local screen format Templates used to provide an interface between you and host computer applications.

Local session The connection of your ASCII terminal to the 3B processor, allowing access to user defined screen layouts (see *Local screen format*) that provide an interface between the user and host applications, and allows the user to exchange data between applications.

Local variable A variable that is defined only for a particular script or subprogram.

Loop A series of instructions that is executed repeatedly until a given condition is met.

Null string A string of zero length (with no characters). It is represented by two double quotation marks ("").

Offset The number of bytes from a starting point to some other point.

Operand A constant or variable that is acted on by an operator. *Operand* also refers to any argument that follows a command or function.

Operator A character that represents a mathematical (or logical) operation. ESCORT uses arithmetic, relational, and string concatenation operators.

Operator information area An area on the terminal display screen, in which messages to the operator are written by ESCORT. On terminals with 25-line screens, this area is on line 25, extending from column 21 to column 80. On terminals with 24-line screens, this area is on line 24, extending from column 21 to column 80. Also called *Status Line.*

Operator notification command A command used to communicate with the operator of a terminal. WTO is an operator notification command.

Overflow An error that develops when the value returned by an operation is too large for a given register or location.

Parameter An argument passed to a subroutine on a CALL statement.

Parser
The ESCORT component that decodes a program and checks syntax.

Preprocessor command
A command that requests an action before program execution. **COPY** is a preprocessor command.

Presentation space
See *Screen buffers.*

Primary Attributes
Four arguments used in a **FIELD** statement that define basic screen characteristics similar to those found in IBM synchronous host screen formats.

Program control command
A command that changes the path of program execution. **BREAK** is a program control command.

Relational operator
A character (such as =) that represents a comparison of two values.

Reserved word
A word used in ESCORT for a special purpose, such as a command or function name. Reserved words cannot be used as names for variables, labels, programs, or scripts.

SDLI card
Synchronous Data Link Interface card. A card installed in the 3B5 or 3B15 processor that provides synchronous communications ability between host computers and terminals.

Screen buffers Buffers maintained by ESCORT which contain the last refreshed screen image for each host and local session. The active session screen buffer can be accessed with the system global variable named *SCREEN*. Also called *Presentation space*.

Screen field variable See *Field variable*.

Script A program written in ESCORT. *Script* also means a subroutine labeled with a script name.

Script mode This mode executes an ESCORT script.

Simulator The ESCORT component that consists of a parser and an interpreter. The simulator checks for correct syntax and executes the parsed code.

Statement An instruction to the computer to perform some sequence of operations. A statement consists of a command or function and its operands.

Status line See *Operator information area*.

String A sequence of characters or words. ESCORT uses string constants, string variables, and string array variables. String constants are enclosed in double quotation marks. String variables are declared in a **CHAR** statement, such as *CHAR (20) name*.

String concatenation operator A symbol that links a series of string operands. The plus sign (+) is used as a string concatenation operator in ESCORT.

Terminal keyboard command A command that simulates the action of a given keyboard function. For example, **CLEAR** is a terminal keyboard command.

Tutorial mode This ESCORT feature provides the ability to perform edit checks on data entered before sending data to the host. It also allows you to set up on-line training sessions.

Variable A symbol used to represent a value. There are five types of variables in ESCORT: integer, integer array, string, string array, and field.

Window A display area on a screen. Windows are defined in ESCORT by a **WINDOW** statement.

Index

Index

308-380
Issue 1