Guardian User's Guide

Abstract

This guide provides basic information about the programs and utilities that are used most often in the Guardian environment by general system or application users. It also provides more detailed procedures for system operations, management, and maintenance tasks useful to both beginning and experienced Compaq *NonStop*™ Kernel operating system users.

Product Version

N.A.

Supported Releases

This manual supports G06.05 and all subsequent G-series releases, and D48.00 and all subsequent D-series releases, until otherwise indicated in a new edition.

Part Number	Published
425266-001	August 2000

Document History

Part Number	Product Version	Published
425266-001	N.A.	August 2000
421957-001	N.A.	June 1999
142477	N.A.	December 1998
117897	N.A.	July 1995
089808	N.A.	January 1993

Ordering Information

For manual ordering information: domestic U.S. customers, call 1-800-243-6886; international customers, contact your local sales representative.

Document Disclaimer

Information contained in a manual is subject to change without notice. Please check with your authorized representative to make sure you have the most recent information.

Export Statement

Export of the information contained in this manual may require authorization from the U.S. Department of Commerce.

Examples

Examples and sample programs are for illustration only and may not be suited for your particular purpose. The inclusion of examples and sample programs in the documentation does not warrant, guarantee, or make any representations regarding the use or the results of the use of any examples or sample programs in any documentation. You should verify the applicability of any example or sample program before placing the software into productive use.

U.S. Government Customers

FOR U.S. GOVERNMENT CUSTOMERS REGARDING THIS DOCUMENTATION AND THE ASSOCIATED SOFTWARE:

These notices shall be marked on any reproduction of this data, in whole or in part.

NOTICE: Notwithstanding any other lease or license that may pertain to, or accompany the delivery of, this computer software, the rights of the Government regarding its use, reproduction and disclosure are as set forth in Section 52.227-19 of the FARS Computer Software—Restricted Rights clause.

RESTRICTED RIGHTS NOTICE: Use, duplication, or disclosure by the Government is subject to the restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 52.227-7013.

RESTRICTED RIGHTS LEGEND: Use, duplication or disclosure by the Government is subject to restrictions as set forth in paragraph (b)(3)(B) of the rights in Technical Data and Computer Software clause in DAR 7-104.9(a). This computer software is submitted with “restricted rights.” Use, duplication or disclosure is subject to the restrictions as set forth in NASA FAR SUP 18-52 227-79 (April 1985) “Commercial Computer Software—Restricted Rights (April 1985).” If the contract contains the Clause at 18-52 227-74 “Rights in Data General” then the “Alternate III” clause applies.

U.S. Government Users Restricted Rights — Use, duplication or disclosure restricted by GSA ADP Schedule Contract.

Guardian User's Guide

Glossary | Index | Figures | Tables

2. Getting Started With TACL (continued)

3. Managing Files With TACL

3. Managing Files With TACL (continued)

4. Starting and Controlling Processes With TACL

5. Defining Function Keys and Writing Macros

5. Defining Function Keys and Writing Macros (continued)

6. Creating and Using DEFINEs

Part II. Managing Files Using the File Utility Program (FUP)

7. Using FUP for Basic File Management

7. Using FUP for Basic File Management (continued)

8. Using FUP for Advanced File Management

Part III. Managing Disk and Tape Processes

9. Performing Routine Disk Operations

9. Performing Routine Disk Operations (continued)

10. Using Labeled Tapes

10. Using Labeled Tapes (continued)

11. Backing Up and Restoring Disk Information

Part IV. Using the Spooler and Its Utilities

12. Introduction to the Spooler

13. Managing Your Spooler Jobs Using Peruse

13. Managing Your Spooler Jobs Using Peruse (continued)

14. Performing Routine Spooler Operations Using Spoolcom

14. Performing Routine Spooler Operations Using Spoolcom (continued)

15. Managing the Spooler Using Spoolcom

15. Managing the Spooler Using Spoolcom (continued)

Part V. Security Features and Other Guardian Utilities

16. Managing Users and Security

16. Managing Users and Security (continued)

17. Monitoring Event Messages

17. Monitoring Event Messages (continued)

18. Displaying Version and System Information

19. Monitoring Hardware Components

19. Monitoring Hardware Components (continued)

A. Problem Solving Techniques

Glossary

Index

Figures

Figures (continued)

Tables

Tables (continued)

What's New in This Guide

Manual Information

Abstract

This guide provides basic information about the programs and utilities that are used most often in the Guardian environment by general system or application users. It also provides more detailed procedures for system operations, management, and maintenance tasks useful to both beginning and experienced Compaq *NonStop*™ Kernel operating system users.

Product Version

N.A.

Supported Releases

This manual supports G06.05 and all subsequent G-series releases, and D48.00 and all subsequent D-series releases, until otherwise indicated in a new edition.

Part Number	Published
425266-001	August 2000

Document History

Part Number	Product Version	Published
425266-001	N.A.	August 2000
421957-001	N.A.	June 1999
142477	N.A.	December 1998
117897	N.A.	July 1995
089808	N.A.	January 1993

New and Changed Information

The G06.09 edition of this guide contains these changes:

- The new Backup and Restore command options NOSQLDATA, REMOTEIOSIZE, and SQLTAPEPARTARRAY were added in Section 11, Backing Up and Restoring Disk Information.
- File and process access information was updated for several procedures and examples in Section 16, Managing Users and Security.

The G06.05 edition of this guide (previously known as the *Guardian 90 Operating System User's Guide*) contained these changes:

- Information was added from the former *Guardian System Operations Guide*, adding several new sections to this guide and expanding on existing sections.
- Updated display information was added for the Getting File Information (FILEINFO Command) on page 3-6.
- New parameters for the FUP SET command were added to Table 8-1, Parameters of the FUP SET Command, on page 8-3.
- Y2K-compliant examples were added and information field descriptions were updated for VPROC to Section 18, Displaying Version and System Information.
- A description of using SYSINFO for displaying basic information about local and remote systems was added to Section 18, Displaying Version and System Information.
- The section about the Disk Space Analysis Program (DSAP) was incorporated into Section 9, Performing Routine Disk Operations.
- Updated the Subsystem Control Facility (SCF) commands and descriptions in Table 9-1, SCF Command Summary, on page 9-1.
- C-series references were removed where appropriate.

About This Guide

This guide provides introductory information and task-oriented instructions for using the Compaq Tandem Advanced Command Language (TACL) and various Guardian environment utilities. The utilities and procedures described in this guide include many of the more common operations tasks that users will need to perform on a system running the Compaq *NonStop*™ Kernel operating system.

What's in This Guide?

Section	Describes...
Section 1, Introduction to Guardian System Operations	The concept of performing operations tasks on a NonStop™ Kernel system in the Guardian environment, and provides a checklist of routine tasks.
Part I. Using the Compaq Tandem Advanced Command Language (TACL)	
Section 2, Getting Started With TACL	The TACL program as a command interpreter and describes how to access the NonStop™ Kernel system, including logging on and off and entering TACL commands. It also introduces Safeguard, a security program, and describes how to log on to a terminal controlled by Safeguard and how to change a password.
Section 3, Managing Files With TACL	Using the TACL program to manage disk files.
Section 4, Starting and Controlling Processes With TACL	Using the TACL program to manage processes that you start.
Section 5, Defining Function Keys and Writing Macros	How to define function keys and write simple TACL macros.
Section 6, Creating and Using DEFINEs	Using DEFINEs, which are sets of attributes and values, to pass information to processes you start.
Part II. Managing Files Using the File Utility Program (FUP)	
Section 7, Using FUP for Basic File Management	FUP, and some of its basic features (renaming, duplicating, and deleting files).
Section 8, Using FUP for Advanced File Management	Using FUP to create and manage disk files.
Part III. Managing Disk and Tape Processes	
Section 9, Performing Routine Disk Operations	How to perform operations relating to disks, such as checking the status of disks, and bringing up and taking down disks.
Section 10, Using Labeled Tapes	Using labeled and unlabeled tapes for systems that have labeled-tape processing.
Section 11, Backing Up and Restoring Disk Information	Using Backup and Restore to copy files between disk and tape.

Section	Describes...
Part IV. Using the Spooler and Its Utilities	
Section 12, Introduction to the Spooler	The Spooler subsystem.
Section 13, Managing Your Spooler Jobs Using Peruse	Using Peruse to manage spooler jobs and spooler job files.
Section 14, Performing Routine Spooler Operations Using Spoolcom	Using Spoolcom, and explains how individual users can check the status of spooler components and change attributes of their spooler jobs.
Section 15, Managing the Spooler Using Spoolcom	Using Spoolcom to manage all spooler collector processes, print processes, print devices, and locations by adding and deleting them from the spooler and by displaying and modifying attributes associated with these processes.
Part V. Security Features and Other Guardian Utilities	
Section 16, Managing Users and Security	How to implement security features using FUP and the TACL program.
Section 17, Monitoring Event Messages	Operator messages and how to interpret them, and mentions several of the programs you can use to monitor them.
Section 18, Displaying Version and System Information	The VPROC and SYSINFO utilities, which provide file and system information, respectively.
Section 19, Monitoring Hardware Components	How to perform system monitoring tasks such as checking the status of system hardware components and key applications.
Appendix A, Problem Solving Techniques	A process you can use to identify, track, and resolve problems that can occur with NonStop™ Kernel systems.

Who Should Use This Guide?

This guide provides introductory information for beginning users of NonStop™ Kernel systems, and detailed procedures for common Guardian-based tasks that are useful to both new and experienced users.

This guide addresses the needs of general system users – users who log on to a NonStop™ Kernel system to run an application program such as electronic mail or a text editor. It also addresses more involved tasks dealing with system maintenance, operations, and troubleshooting, that is useful to experienced operators, system managers, and group managers.

Where Else Can You Find Information?

Before reading this guide, you should be familiar with the NonStop™ Kernel concept of files and processes, and understand the concept of a command-interpreter interface. For information on these concepts and the NonStop™ Kernel utility programs, see:

Manual	Describes
Introduction to Tandem NonStop Systems	A general introduction to NonStop™ Kernel systems and online transaction processing (OLTP)
Guardian Disk and Tape Utilities Reference Manual	Command syntax and error messages for Backup, Restore, Backcopy, DSAP, DCOM, and Tapecom
File Utility Program (FUP) Reference Manual	FUP command syntax and error messages
Spooler Utilities Reference Manual	The Spooler and the command syntax and error messages for Peruse and Spoolcom
TACL Reference Manual	Syntax, operation, results, and error messages for TACL commands and functions
Safeguard User's Guide	Greater detail for logging on at a terminal controlled by Safeguard, the subsystem that supplements system security features

Your Comments Invited

After using this guide, please take a moment to send us your comments. You can do this by returning a Reader Comment Card or by sending an Internet mail message.

A Reader Comment Card is located at the back of printed manuals and as a separate file on the TIM disc. You can either FAX or mail the card to us. The FAX number and mailing address are provided on the card.

Also provided on the Reader Comment Card is an Internet mail address. When you send an Internet mail message to us, we immediately acknowledge receipt of your message. A detailed response to your message is sent as soon as possible. Be sure to include your name, company name, address, and phone number in your message. If your comments are specific to a particular manual, also include the part number and title of the manual.

Many of the improvements you see in Compaq manuals are a result of suggestions from our customers. Please take this opportunity to help us improve future manuals.

1

Introduction to Guardian System Operations

Most Compaq *NonStop*™ Kernel operating system users regularly interact with their systems in the Guardian environment. The Guardian environment contains many programs, tools, and utilities that perform most of the core tasks involved in operating a NonStop™ Kernel system, including:

- Basic file management
- Security management
- Working with the print spooler
- Tape operations
- Backups
- Event monitoring

This guide explains the fundamentals of many Guardian tools, describes many common tasks in these areas that are usually performed on a routine basis, and references where to get more information for advanced usage.

Some of the tasks in this guide, such as working with the print spooler and labeled tapes, describe software procedures that work with Compaq *NonStop*™ Himalaya server hardware. Hardware procedures are not discussed in this guide, except for an introduction to some common hardware monitoring.

Most NonStop™ Kernel system users will use Guardian at some point, and this guide contains procedures that apply to most users. But most of the procedures in this guide are usually carried out by the most common Guardian users: system operators.

This section introduces you to the system operations fundamentals and tasks described in this guide:

Common Guardian Operations Tasks

Tasks that are frequently performed in the normal day-to-day operation and maintenance of a NonStop™ Kernel system are referred to as “operator tasks.” In this guide, the terms “system operator” or “operator” apply broadly to anyone performing operator tasks. Sometimes operators will only perform basic tasks, not those for which a super-group user ID (255, *n*) is required. However, this is not always the case. Also, super-group users sometimes perform basic operator tasks. For these reasons, many common tasks requiring a super-group user ID are also included in this guide.

System operators have many areas of responsibility and perform many tasks. There are certain fundamentals to know in the Guardian environment that will help you carry out the most basic operator tasks, including:

Task	Section
Routine system access	Section 2, Getting Started With TACL
Working with files	Section 3, Managing Files With TACL
Routine process operations	Section 4, Starting and Controlling Processes With TACL
Routine user customization	Section 5, Defining Function Keys and Writing Macros
Creating DEFINEs	Section 6, Creating and Using DEFINEs
Routine file management	Section 7, Using FUP for Basic File Management
Advanced file management	Section 8, Using FUP for Advanced File Management

Whether your job requires you to staff a help desk, perform tape backups, or serve as a lead operator or manager of other operators at your site, system operations centers around the day-to-day operation of a NonStop™ Kernel system, including:

Task	Section
Routine disk operations	Section 9, Performing Routine Disk Operations
Routine tape operations	Section 10, Using Labeled Tapes
Running backups	Section 11, Backing Up and Restoring Disk Information
Spooler job operations	Section 13, Managing Your Spooler Jobs Using Peruse
Routine spooler operations	Section 14, Performing Routine Spooler Operations Using Spoolcom
Spooler management operations	Section 15, Managing the Spooler Using Spoolcom
Supporting users of your system	Section 16, Managing Users and Security
Monitoring operator messages	Section 17, Monitoring Event Messages
Viewing file or system data	Section 18, Displaying Version and System Information
Monitoring system hardware and software status	Section 19, Monitoring Hardware Components
Identifying and solving system problems	Appendix A, Problem Solving Techniques

Working With a Daily Check List

Regardless of which shift you work, you need to regularly check certain areas of your hardware and software environment.

A good method for ensuring that certain areas of your operations environment are checked is to develop a check list for yourself and other operators to follow.

Table 1-1 provides an example of the areas you should check at the beginning of any shift, a summary of the tasks to perform for each area, and where in this guide you can find more detailed instructions.

Table 1-1. Daily Tasks Check List

General Tasks	Specific Tasks	More Information
Check disks	Use the SCF STATUS DISK command and DSAP	Section 9, Performing Routine Disk Operations
Check tape drives	Use the SCF STATUS TAPE and MEDIACOM STATUS TAPEDRIVE commands	Section 10, Using Labeled Tapes
Check printers and respond to any spooler problems	Use the SPOOLCOM DEV command	Section 14, Performing Routine Spooler Operations Using Spoolcom
Check for messages from system users	Check telephone, fax, electronic mail, and any other messages	Section 16, Managing Users and Security
Check operator messages	Use the TMS EMS Event Viewer, a printing distributor, or other application	Section 17, Monitoring Event Messages
Check system status, (terminals, processors, communication lines, key applications, and system processes)	Use SCF, Pathway, TMF, or the TACL PPD command	Section 19, Monitoring Hardware Components

Your System Might Be Protected

Safeguard, an optional NonStop™ Kernel security software product, might be installed on your system, extending the operating system's security features by:

- Protecting additional system resources
- Providing more flexibility in sharing those resources
- Recording security-relevant events for later analysis

If Safeguard is on your system, you might experience some differences when performing the tasks described in this guide, such as receiving a security violation (error code 48). However, the Safeguard product is flexible and can be configured for different situations. Consult the *Security Management Guide* or your security administrator or system manager if you have questions about your system or network.

Automating Routine Monitoring Tasks

Automating routine tasks and procedures helps save you time and reduces the possibility of errors. Most of the tasks in Table 1-1 can be automated using command files and TACL macros or routines. Automation examples for many of the tasks in this guide are included with the task descriptions.

Hardware operations manuals such as the *Himalaya S-Series Operations Guide* contain specific examples for automating system startup operations.

Example Check List

This example shows a form you might use to standardize your routine start-of-shift monitoring tasks:

Task	Operator's name	Date & time	Notes and questions
Check phone messages			
Check faxes			
Check e-mail			
Check shift log			
Check TSM EMS Event Messages			
Check status of terminals			
Check comm. lines			
Check TMF status			
Check Pathway status			
Check disks			
Check tape drives			
Check CPUs			
Check printers			
Check spooler supervisor and collector processes			

Part I. Using the Compaq Tandem Advanced Command Language (TACL)

This part of the guide contains information about using TACL as the interface between you and your Compaq *NonStop*™ Kernel operating system:

- Section 2, Getting Started With TACL
- Section 3, Managing Files With TACL
- Section 4, Starting and Controlling Processes With TACL
- Section 5, Defining Function Keys and Writing Macros
- Section 6, Creating and Using DEFINEs

Part I. Using the Compaq Tandem Advanced Command Language (TACL)

2 Getting Started With TACL

The Compaq Tandem Advanced Command Language (TACL) program, the command interpreter for the Guardian environment, is the primary interface between you and the Compaq *NonStop*™ Kernel operating system.

The TACL program lets you designate special function keys and create macros:

- Assign an alias, or alternate name, to a TACL command or an application. You can execute the command or application by simply entering the alias name.
- Write a macro to execute TACL commands or run an application. A macro is a named sequence of one or more TACL commands stored in an EDIT file. Entering the macro name invokes the command sequence or application.
- Define your function keys to execute TACL commands, invoke macro and alias definitions, and run applications.
- Write programming routines to perform complex operations.

See Section 5, Defining Function Keys and Writing Macros, for more information.

The TACL program can also be used as a programming language, but this is not described in this guide. See the *TACL Reference Manual* and the *TACL Programming Guide* for more information.

The Safeguard subsystem is a group of software security programs that supplements the system security features that is commonly used to secure user logons to TACL sessions.

Using TACL as a Command Interpreter

The TACL program is most often used interactively as a command interpreter:

1. After system startup, your system manager or operator starts a TACL process from the TACL program file for each terminal, including yours, connected to the system. (The TACL program file is on the system disk $SYSTEM.SYS*nn*.TACL, where *nn* is a two-digit number.)
2. You log on to the TACL process at your terminal to access your system (see Logging On With TACL on page 2-3). If you are connected on a network, you can also start a TACL process on another system (see Section 4, Starting and Controlling Processes With TACL).

   On systems running Safeguard software, your system manager can optionally specify that some or all terminals be controlled by Safeguard. At these terminals, the logon procedure differs from the standard TACL logon. A successful Safeguard logon results in a TACL prompt only if your system administrator has configured the system to do so (see Logging On With Safeguard on page 2-9).
3. After you log on at your terminal, you can enter TACL commands at your terminal keyboard. As a command interpreter, the TACL program lets you enter TACL commands, run system utilities (such as TEDIT or Peruse), and run any application that you might need to do your job.

Entering TACL Commands

You can enter interactive TACL commands only when the TACL prompt (>) appears on your terminal screen, which indicates that TACL is waiting for you to enter a command. Type your command after the prompt, then press Return.

To use the TIME command to make the TACL program display the system date and time:

```
5> TIME
```

You must end each TACL command line by pressing Return. Examples in this guide assume that you press Return at the end of each command line.

Most TACL commands are short (fewer than 80 characters). They can, however, contain up to 239 characters on one line. To enter commands longer than 239 characters:

1. End each command line (except the last line) with an ampersand (&).

   The TACL program redisplays its prompt with the same line number and an ampersand at the beginning of the continuation line.
2. Continue entering your command after this ampersand on the continuation line.

For example, these two commands produce the same results:

```
10> TAL / IN $MANUF.MYSUB.MYSRCE, OUT $LP / $MANUF.MYSUB.MYOBJ

10> TAL / IN $MANUF.MYSUB.MYSRCE, OUT $LP / $MAN&
10> &UF.MYSUB.MYOBJ
```

These examples show several simple, common ways to enter a TACL command. For information about entering more than one TACL command per line and other ways to continue TACL commands over several lines, see the *TACL Reference Manual*.

Logging On With TACL

To access your system and establish communication with a TACL process, use the TACL LOGON command. To log on at a terminal controlled by Safeguard, see Logging On With Safeguard on page 2-9.

To log on, you need a user name and user ID, which are usually assigned to new users by your group or system manager.

Your user name has two parts, separated by a period: your group name and your individual name within your group. For example, the user name of a user named Stein in the Support group would be:

```
SUPPORT.STEIN
```

Your user ID is your group number and your individual number in your group, separated by a comma. For example, if user SUPPORT.STEIN is user number 66 in group number 6, her user number is:

```
6,66
```

Each user can have a logon password — a string of characters that you must enter to access the system. You can select and change your own password on most systems (see Changing Your Password on page 2-8). Using a logon password prevents anyone else from logging on as you. If you do not use a password, anyone can log on as you by simply entering your user name in a LOGON command.

Passwords are case sensitive, which means that lowercase and uppercase letters are recognized as different characters and you must enter them appropriately.

If you have a user name and ID, but no password, you log on by entering your user name in a LOGON command. When the TACL program prompts for a password, press Return.

The TACL program provides two methods for logging on to a system using a password:

- Blind password logon feature
- Full logon feature

Note. Some systems do not require a password for users. Other systems might not allow the full logon feature or user IDs in a LOGON command. If you are uncertain about your system, ask your group or system manager.

Blind Password Logon

The blind password logon feature lets you log on without displaying your password.

1. For SUPPORT.STEIN to log on, Stein enters this at the initial TACL prompt:

```
TACL 1> LOGON SUPPORT.STEIN
```

   When Stein presses Return at the end of this line, the TACL program requests the password:

```
TACL 1> LOGON SUPPORT.STEIN
Password:
```

2. At the prompt, Stein enters the password exactly. The typed password, however, is not displayed on the screen and thus remains secret.

You can log on with either your user name or your user ID. For example, user SUPPORT.STEIN can also enter:

```
TACL 1> LOGON 6,66
```

Some systems are configured to allow only the blind logon feature, which means the full-logon feature would not be allowed.

Full Logon

The full logon feature lets you enter your password at the same time you enter your user name or ID. This is faster, but your password is visible to anyone watching you. For example, using the same user name from the previous example, with the password ABT (all uppercase), Stein logs on to the system using the full logon feature:

```
TACL 1> LOGON SUPPORT.STEIN,ABT
```

The password is typed after the user name and is separated from it by a comma. When Stein presses Return at the end of this line, the TACL program lets Stein access the system (provided the user name and password are valid).

You can also use the full logon feature with your user ID:

```
TACL 1> LOGON 6,66,ABT
```

Logon Mistakes

If you make a mistake entering your user name, user ID, or password, the TACL program displays this message:

```
*ERROR* Invalid user name or password
```

If you make three unsuccessful logon attempts, the TACL program ignores any attempts to log on from your terminal for one minute. All subsequent logon failures also cause this delay. (Your system manager might set the delay to be longer than one minute.)

A Successful Logon

After you have logged on, the TACL program displays a message similar to this and issues your first command prompt.

```
Good Morning. Welcome to \MEL
TACL (T9205D10 - 08JUN92), Operating System D10
COPYRIGHT TANDEM COMPUTERS INCORPORATED 1985, 1987, 1988, 1989, 1991, 1992
CPU 1, backup process in CPU 2
February 20, 1992 9:15:23
(Invoking $SYSTEM.SYSTEM.TACLLOCL)
(Invoking $GERT.STEIN.TACLCSTM)
1>
```

This message contains:

- A system greeting.
- The TACL program banner, which tells you the product number, and release version and date of the TACL program, and the operating system release version.
- The number of the CPU that is running your TACL process and your backup TACL process, if you have one.
- The current system date and time.
- Statements telling you that your TACL process has invoked two files for customizing your TACL environment—TACLLOCL and TACLCSTM (see Section 5, Defining Function Keys and Writing Macros).
- Your first TACL command prompt, which is a greater-than sign (>) preceded by a command line number.

You are now ready to enter TACL commands.

While logged on, you or another user can log on without your having to log off. The user ID currently logged on is automatically logged off when another LOGON command is entered. For example:

```
2> LOGON SUPPORT.ALICE
Password:
```

When SUPPORT.ALICE logs on, SUPPORT.STEIN is automatically logged off. The user ID and current defaults are changed to those of SUPPORT.ALICE. However, the TACL program retains the currently defined variables for STEIN (such as the macros and function-key definitions, if set).

Logging Off With TACL

After you log off, processes that you started continue to run. Therefore, before you log off, you should stop any processes that you no longer need (see Section 4, Starting and Controlling Processes With TACL).

To end your TACL session, use the LOGOFF command:

```
30> LOGOFF
```

When you log off, the TACL program clears your screen (if your terminal is a Compaq model 652X or 653X or if you are running a terminal emulator such as PCT) and returns the initial TACL prompt (TACL 1>). If the TACL program does not clear your screen, use the CLEAR option with the LOGOFF command by entering `LOGOFF CLEAR`.

After you log off, any ASSIGN, PARAM, and DEFINE commands are lost. However, your variables, such as alias and macro definitions, are not lost unless your segment was reset. See the LOGOFF command in the *TACL Reference Manual* for more information.

Accessing Other Systems

When NonStop™ Kernel systems form a network using Expand, access to a file can be restricted to users on the local system where the file resides, or access can be allowed for users on any system in the network. (Safeguard can secure a file so that only specific individuals, either locally or on the network, can access that file.)

To access a file available only to local users, you must be logged onto the local system. To log onto a system other than the one where your current TACL process is running, you must:

1. Be established as a user on the remote system, having the same user name and ID on both systems, and having remote passwords set up between your local and remote systems.
2. Start a remote TACL process in that system.

A remote password is not a password that you use when you log on. It is an indicator that the remote system checks when you attempt to access that system.

For example, if your local system is \ABT, and you want access to \FERN and \HERST, you must be added as a user to those systems with identical remote passwords on all three systems.

Table 2-1. Sample Remote Passwords

System \ABT	System \FERN	System \HERST
\ABT AB	\ABT AB	\ABT AB
\FERN FE	\FERN FE	\FERN FE
\HERST HE	\HERST HE	\HERST HE

Establishing Remote Passwords

To establish a remote password, you (or a system manager) must log onto each system on which you want to establish a remote password, and enter a REMOTEPASSWORD command, which invokes the RPASSWRD utility. A remote password can contain from one to eight alphanumeric, nonblank characters, and is case-sensitive.

This example establishes passwords FE, HE, and AB for systems \FERN, \HERST, and \ABT. Log on and enter these commands at each system:

```
16> REMOTEPASSWORD \FERN, FE
THE \FERN REMOTE PASSWORD FOR SUPPORT.STEIN (8,56) HAS BEEN CHANGED.
17> REMOTEPASSWORD \HERST, HE
THE \HERST REMOTE PASSWORD FOR SUPPORT.STEIN (8,56) HAS BEEN CHANGED.
18> REMOTEPASSWORD \ABT, AB
THE \ABT REMOTE PASSWORD FOR SUPPORT.STEIN (8,56) HAS BEEN CHANGED.
```

Starting and Quitting a Remote TACL Process

Once you are an established user on all the systems you want to access and you have all the necessary remote passwords, you can start a TACL process on any of those systems. Enter a command that specifies the system followed by a period and the TACL program file name.

For example, if your local system is part of a network that includes the system \HERST, you can start a TACL process on \HERST by entering:

```
19> \HERST.TACL
TACL 1>
```

The TACL program returns the initial TACL prompt, and you can log onto \HERST.

A remote TACL process started this way does not have a backup process. If you want the remote TACL process to run as a NonStop™ process pair instead, enter:

```
19> \HERST.TACL / NAME, CPU 1 / 2
TACL 1>
```

The operating system assigns a name to the process pair and starts the process in CPU 1 with a backup process in CPU 2.

If you don’t know the CPU numbers for the remote system, start the primary process:

```
19> \HERST.TACL / NAME /
TACL 1>
```

Then, after you are logged on, determine the CPU numbers for the remote system and issue a BACKUPCPU command:

```
3> BACKUPCPU 4
```

You can also use the SYSTEM or VOLUME command to change your current default system, then you can start a remote TACL process without specifying the system. For example, these commands start a TACL process on the system \HERST:

```
19> SYSTEM \HERST
20> TACL
TACL 1>
```

To quit the remote TACL process, enter:

```
5> EXIT
Are you sure you want to stop this TACL (\HERST.$Z100)?
```

Enter YES (or Y) to stop the remote process. Stopping the remote TACL process returns you to the TACL process on your local system. If you do not want to stop the process, enter any other character, or press Return.

Do not stop your local TACL process unless it is necessary, because that makes it impossible for you to access your terminal.

Changing Your Password

A logon password can contain from one to eight letters, numbers, control characters, or other characters, but must not contain:

- Blanks or commas
- CTRL-A
- Certain other control characters (on some terminals; check your terminal manual)

To change a logon password , or to choose a password if you don't have one:

1. Log on to your terminal.
2. Change the password (in this example, to BASKET) by entering:

```
4> PASSWORD BASKET
```

The Password program responds with this message:

```
THE PASSWORD FOR USER (006,066) HAS BEEN CHANGED.
```

To delete your password, enter a PASSWORD command without including a new password:

```
6> PASSWORD
THE PASSWORD FOR USER (006,066) HAS BEEN CHANGED.
```

Note. On some systems, the Password program is set up to require that you enter a blind password (one that is not displayed on your screen) and that you verify your old password before you can change it. If you have problems running Password, check with your group or system manager for your system's requirements.

Logging On With Safeguard

The Safeguard subsystem is a group of programs that supplements the security features of the system. If Safeguard software is installed on your system, your system manager can specify that some, none, or all of the terminals on your system be controlled by Safeguard.

The Safeguard Logon Prompt

You can tell whether a terminal is controlled by Safeguard software because the logon prompt is different from the TACL logon prompt:

```
SAFEGUARD 1>
```

If this prompt does not appear, the terminal is not controlled by Safeguard software and you should follow the instructions in Logging On With TACL on page 2-3.

The Safeguard logon prompt accepts only two commands: LOGON and TIME. If you enter any other command, Safeguard displays this message:

```
Expecting: LOGON or TIME
```

The TIME Command

The TIME command displays the current date and time, for example:

```
SAFEGUARD 1> TIME
15 FEB 1992, 12:46:21
```

The Safeguard LOGON Command

When a terminal is controlled by Safeguard, you must use the Safeguard LOGON command to access your system.

To terminate the LOGON command during entry, press CTRL-Y.

The LOGON command can accept your user name and password in several different formats, depending on how the Safeguard software is configured.

Note. Systems running Safeguard software can be configured several different ways. The examples in this section represent a standard configuration. If you are uncertain about your system, see the *Safeguard User's Guide* or ask your group or system manager.

Blind Password Logon

To enhance system security, blind password logon is standard in Safeguard system configurations. Your password is not displayed on the screen when you type it, and you must log on with your user name instead of your user ID). For example:

```
SAFEGUARD 1> LOGON support.stein
```

This example shows the complete Safeguard logon for SUPPORT.STEIN. The user's password (ABT) is shown here although it is not displayed on the screen when entered:

```
SAFEGUARD 1> LOGON support.stein
Password: ABT
*WARNING* Password Expires 20 May 1992, 12:00
Last Logon: 14 May 1992, 08:02:23
Good Morning. Welcome to \MEL
```

This logon display contains messages that:

- Tell you when your current password will expire
- Indicate the date and time of the last successful logon for this user name
- Greets you and typically includes the name of the system being accessed

Changing an Unexpired Password

1. During the normal logon dialog, type a comma (,) at the end of your password.

   Safeguard prompts you for a new password.

2. Enter your new password.

   Safeguard prompts you to retype the new password to verify it.

3. Retype your new password.

This example shows the new password, but it is not displayed on your screen when you type it:

```
SAFEGUARD 1> LOGON support.stein
Password: ABT,
Enter new password: Basket
Reenter new password: Basket
The password for SUPPORT.STEIN has been changed.
Last Logon: 14 May 1991, 08:02:23
Good Morning. Welcome to \MEL
```

Safeguard passwords are case-sensitive. In this example, each time SUPPORT.STEIN logs on, the letter B in the new password must be capitalized.

Changing Your Password When Already Logged In

If the Password program is available on your system, you can use it to change your password after a successful Safeguard logon. See Changing Your Password on page 2-8.

Changing an Expired Password

You might have a grace period during which you can change your password after it expires. If you have ignored warnings of an upcoming expiration date and have let your password expire, you can change your password during this grace period.

This example shows the new password, but it is not displayed on your screen when you type it:

```
SAFEGUARD 1> LOGON support.stein
Password: ABT
Password expired
Enter new password: Basket
Reenter new password: Basket
The password for SUPPORT.STEIN has been changed.
Last Logon: 14 May 1999, 08:02:23
Good Morning. Welcome to \MEL
```

Because Safeguard can be configured in different ways, this option might not be available on your system. Check with your group or system manager if you have questions about changing an expired password.

Logging On to a Remote System

To log on to a remote system, your user ID must have been added to the remote system and you must have remote passwords established to allow network access. See Establishing Remote Passwords on page 2-7.

To access a remote system on your network, you must use the Safeguard LOGON program.

1. Log on to your local system.
2. Log on to the remote system from your local terminal.

   The Safeguard LOGON program initiates the logon prompt from Safeguard software on the remote system.

This example shows how to use the LOGON program to log on to the remote system named \SFO.

```
4> RUN \SFO.LOGON
SAFEGUARD 1>
```

When you receive the Safeguard logon prompt, you log on normally.

Getting TACL Help

The TACL program offers on-screen help for TACL commands:

- When you make a mistake entering a command, the TACL program displays ways to correct your mistake.

  For example, when you misspell the name of a command, the TACL program displays a list of the options it is expecting at that point:

```
7> STATS *, USER
STATS *, USER
^
*ERROR* Name of variable, builtin, or file needed
```

  This message tells you that the TACL program needs the correct name of a variable (such as a TACL command or macro), a built-in (a TACL function or variable that begins with a number sign “#”), or a file (the name of a program file). In this case, STATS is none of these; it is a misspelling of STATUS.

  If you enter the command name correctly but make a mistake in a command option, the TACL program lists the options that are legal for that command. For example:

```
8> STATUS *, USR
STATUS *, USR
          ^
Expecting one of:
DETAIL GMOMJOBID PRI PROG STOP TERM or USER
```

- The F16 function key is the TACL HELP key. If you enter the first part of a command plus a space and then press F16 (without pressing Return), the TACL program lists the possible options for that point in the command.

  This example shows a list of options for the STATUS command. In this example, the user pressed the F16 key after typing STATUS and a space.

```
9> STATUS
status
       ^
Where you typed the helpkey, TACL was expecting /
Or a legal processid or cpu,pin
Or a legal system name
Or a number or an arithmetic expression
(Its value must be between 0 and 15 inclusive)
Or *
Or ,
Or end
```

Displaying User Information

During a TACL session, you might need to get information about yourself or other system users to:

- Specify your TACL process or your home terminal in a subsequent command
- Check your current defaults (volume, subvolume, and security)
- Obtain the user name associated with a specific user ID

Displaying Your Information

To display your current default settings and other information about your current TACL process, use the TACL WHO command:

```
10> WHO
```

The TACL program displays:

```
Home terminal: $TJ3.#33
TACL process: \MEL.$TJ33
Primary CPU: 5 (VLX)  Backup CPU:     4 (VLX)
Default Segment File: $STEIN.#6539
  Pages allocated: 8  Pages Maximum: 1024
  Bytes Used: 13364 (0%)  Bytes Maximum: 2097152
Current volume: $GERT.STEIN
Saved volume: $GERT.STEIN
Userid: 6,66  Username: SUPPORT.STEIN  Security: "UUUU"
```

The WHO command display includes:

Home terminal	Name of the terminal where you are logged on. If you are logged on to a remote system, the terminal name also includes the system name.
TACL process	System and process name of your TACL process.
Primary CPU	CPU number where your primary TACL process is running and the type of processor.
Backup CPU	CPU number where your backup TACL process is running and the type of processor.
Default Segment File	Temporary disk file created by the TACL program to store your variables.
Pages allocated	Number of pages and bytes of memory that the TACL program can use to store information.
Bytes used	Estimate of the amount of memory already used.
Current volume	Names of your current default disk volume and subvolume. Whenever you specify a partially qualified file name, the TACL program uses the current defaults for the omitted parts of the file name.
Saved volume	Names of your saved default disk volume and subvolume. These are the defaults that are in effect each time you log on.

Userid	Your user ID (group number, user number).
Username	Your user name (group name.user name).
Security	Your default file security string.
Default process	Process ID (CPU number and process number) of the last process you started, if that process is still running.

Displaying Information About Other Users

To get information about a user, enter USERS and the user name or user ID:

```
11> USERS SUPPORT.STEIN
GROUP . USER      I.D. #       SECURITY    DEFAULT VOLUMEID
SUPPORT.STEIN     006,066         OOOO      $GERT.STEIN
```

The USERS program displays the user's name, ID, default file security, and default volume. These defaults are described in Changing Your Default Values on page 3-10.

To get information about all the users in the specified group number, use an asterisk (*) with a group number. For example, this command displays information about all members of group 6:

```
13> USERS 6,*
GROUP . USER      I.D. #       SECURITY    DEFAULT VOLUMEID
SUPPORT.STEIN     006,066         UUUU      $GERT.STEIN
SUPPORT.ALICE     006,075         NUNU      $GERT.ALICE
```

To list all users in the specified group, enter the group name followed by a period (.) and an asterisk (*):

```
14> USERS PAYROLL.*
```

To list all users in your own group, enter an asterisk (*):

```
15> USERS *
```

Using Your Command History

While you are logged on, the TACL program keeps a record of the commands you enter in a history buffer. The history buffer retains up to 1000 characters, enough to save a large number of typical TACL commands.

The TACL program has four commands (HISTORY, ?, !, and FC) that let you recall, correct, change, or reexecute a previous command.

Listing Your Previous Commands

Use the HISTORY command to display your previous commands during this TACL session. If you do not specify a number of commands to display, the default is to display your previous ten commands.

To display the last ten commands you entered, enter:

```
> HISTORY
```

Your previous ten commands are displayed:

```
16> HISTORY
7> STATS *, USER
8> STATUS *, USR
9> STATUS
10> WHO
11> USERS SUPPORT.STEIN
12> USERS 6,66
13> USERS 6,*
14> USERS PAYROLL.*
15> USERS *
16> HISTORY
```

If you do not specify a number in the command and have entered fewer than ten commands during this TACL session, HISTORY displays all your commands beginning with number one.

To display any specific number of commands, enter a number in the command:

```
> HISTORY 4
```

In this example, your previous four commands are displayed:

```
17> HISTORY 4
14> USERS PAYROLL.*
15> USERS *
16> HISTORY
17> HISTORY 4
```

Redisplaying a Selected Previous Command

The question mark (?) command redisplays a command from your history. To identify the command line, you can use either the command line number, the relative line number, or a text string that uniquely identifies the command.

To see the command you issued at line 11, enter the ? command and the number 11:

```
18> ? 11
18> USERS SUPPORT.STEIN
```

To see a command that you issued a specific number of lines back, use a relative number with ?. Use a negative number to signal a relative number. For example, -3 indicates that you want to see the command you issued three lines back.

To display a command that begins with a text string, use ? with that string. The TACL program searches for the most recent command that begins with the string. For example, if you know you issued a PASSWORD command but can't remember the line number, enter ? and enough characters to identify the command:

```
18> ? PAS
18> PASSWORD
```

To list the last command you entered, enter ? with no number or text string.

Reexecuting a Previous Command

The exclamation point (!) command recalls a specific command like the ? command does, and immediately executes the command.

To see and immediately reexecute the command you entered on line 14, enter:

```
18> ! 14
18> USERS PAYROLL.*
```

You can also use a relative number. For example, to reexecute command line 11 when the current line is 19, enter:

```
19> ! -8
19> USERS SUPPORT.STEIN
```

To reexcute the most recent command that matches a certain text string, include the text string with the ! command. The TACL program searches your command history for the most recent command whose text begins with the same string, then executes it:

```
20> ! US
20> USERS *
```

To reexecute the last command you entered, enter ! without a number or text string.

Changing or Correcting a Previous Command

The FC command allows you to change or correct any command in your command history. To recall a command line:

- To recall the command on the previous line, enter only FC:

```
21> FC
21> USERS *
21..
```

- To recall a command on a specific line, enter FC and the line number:

```
22> FC 13
22> USERS 6,*
22..
```

- To recall a previous command, enter FC and negative number to indicate the position of the command in the command history:

```
23> FC -9
23> USERS PAYROLL.*
23..
```

- To recall a command that begins with a specific text string, enter FC and the string:

```
24> FC USE
24> USERS *
24..
```

After you select the command, make your changes. The command editing prompt (`..`) appears below the line that you recalled. On the prompt line, use the space bar and the backspace key to position the cursor under the text that you want to change. Do not use the arrow keys to move the cursor.

Using the Editing Characters D, I, and R

The FC command accepts three command-editing characters:

D (or d)	Deletes the character above the D
I (or i)	Inserts the text following the I into the command line (that is, text is inserted in front of the character above the I)
R (or r)	Replaces characters in the command line beginning with the character above the R with the text following the R

You must begin your correction with these editing characters if the first character of the change is I, D, or R. Type the D or R under the character to be deleted or replaced. Type the I under the character that follows the insert position.

After you make changes to the command, press Return. The TACL program then shows you the modified command and prompts you for more changes. You can add more changes at this point, or if the command is correct, press Return again, and the TACL program executes the command.

Spaces typed after the I or R command are part of the text to insert or replace. If you want to make more than one change on a line, end the text string with two slashes (//) and space over to make additional changes as shown below:

```
25> STAUS *> TERM
STAUS *> TERM
^
*ERROR* Name of variable, builtin, or file needed
26> FC
26> STAUS *> TERM
26.. iT//r,
26> STATUS *, TERM
26..
```

In the example, the I command inserts a T between A and U. The two slashes indicate the end of this insertion. The R command with the comma under the > replaces the > with the comma.

After you press Return, the TACL program displays the corrected line and prompts you for any further changes. Because the command is now correct, make no changes before you press Return, and the TACL program executes the modified command.

Replacing and Inserting Text Without Using D, I, or R

You can enter a replacement string under text without using the R command. You can also insert text at the end of the line by typing it in without using the I command. For example:

```
25> STAUS *>
STAUS *>
^

*ERROR* Name of variable, builtin, or file needed
26> FC
26> STAUS *>
26..  TUS *, TERM
26> STATUS *, TERM
26..
```

However, if your proposed replacement or insertion text begins with the letters D, I, or R (lowercase or uppercase), FC considers that letter to be an editing command, and you will not get the results you want. The following example shows an attempt to change the command from VOLUME ALWORK to VOLUME ALINFO by typing INFO under WORK.

```
27> VOLUME ALWORK
28> FC
28> VOLUME ALWORK
28..   INFO
28> VOLUME ALNFOWORK
28..
```

The TACL program, however, interprets the I in INFO as the I command and inserts the string NFO after the L. At this point, rather than try to execute or edit the command, you can enter two forward slashes (//) followed by Return.

The FC command terminates and the TACL prompt returns. All existing changes to the line are discarded:

```
28> VOLUME ALWORK
28..     INFO
28> VOLUME ALNFOWORK
28..//
```

To terminate the FC command, press CTRL/Y.

To change VOLUME ALWORK to VOLUME ALINFO, enter:

```
29> FC
29> VOLUME ALWORK
29..       RINFO
29> VOLUME ALINFO
29..
```

In this case, the R command replaces the string WORK with the string INFO.

Changing or Correcting Multiple-Line Commands

Suppose that earlier in your TACL session, you entered this command:

```
10> RUN $APPS.EMPLOYEE.LISTPGM /IN $DISK88.EMPLOYEE.NAMES,&
10> &OUT $S.#LASER, NOWAIT/
```

To reuse the command to send the output to a different spooler location:

1. Determine the command number with the HISTORY command
2. Recall it with the FC command.

   The TACL program displays the command with the current command number. The TACL program does not display the ampersands and does not necessarily break the command where you originally broke it.

```
30> RUN $APPS.EMPLOYEE.LISTPGM /IN $DISK88.EMPLOYEE.NAMES,O
30> UT $S.#LASER, NOWAIT/
```

3. Change the first line as you would change a single-line command.
4. Move the cursor with the space bar (not the arrow keys) until it wraps around and is under the second line.
5. Change the second line as you would change a single-line command.

In this example, one command was entered on two lines by enclosing the command with square brackets (#INFORMAT is set to TACL).

```
15> [ RUN $APPS.EMPLOYEE.LISTPGM / IN $DISK88.EMPLOYEE.NAMES,
15> OUT $S.#LAZR, NOWAIT / ]
```

To change and reexecute this command to send the output to the spooler location #HOLD:

1. Recall the command with the FC command.

   The TACL program displays the first line of the command:

```
29> FC [R
30> [ RUN $APPS.EMPLOYEE.LISTPGM / IN $DISK88.EMPLOYEE.NAMES,
30..
```

2. If you don’t have any changes for the first line, press Return.

   The TACL program displays the second line with the same line number.

3. Change the spooler location to #HOLD just as you would change a single-line command, and press Return.

```
30> OUT $S.#LAZR, NOWAIT / ]
30..        HOLD
```

4. The TACL program again displays each line for you to confirm your change.
5. Make additional changes, as needed. Press Return after each line.

   After the last line, the TACL program executes the changed command.

3 Managing Files With TACL

Use the TACL program to perform basic file management tasks with your disk files:

Selected TACL commands are described in this section. For syntax and reference information about all TACL commands and programs, see the *TACL Reference Manual*.

Introduction to Files in Guardian

On a Compaq *NonStop*™ Kernel system, a file on the Guardian operating system can be:

- Disk files, which can contain data, code, or text
- Nondisk devices, such as terminals, printers, or tape drives
- Processes (programs that are running)

You always refer to a file by its file name. For disk files, the user or process who creates the file gives it a name. Nondisk devices have assigned names in the system. Processes can be named by their creator or can be assigned a name by the operating system.

A disk file name includes the file's node, volume, subvolume, and file identifier. File names for devices other than disks and processes begin with a dollar sign ($) followed by one to five alphanumeric characters. For example:

```
$DAISY (a printer name)
$CMON (a process name)
$CD55 (a process name)
\WEST.$DATA.SEND.MSGS (a disk file name)
```

Types of Disk Files

You can create disk files to store databases, coded programs, or text. For most uses, you should create the specific type of file that best suits your purpose. Even if you do not create your own files, you might work with various types of files and need to perform FUP operations on those files.

Enscribe, the NonStop™ Kernel database record manager, supports four disk file types:

- Unstructured files
- Key-sequenced files
- Entry-sequenced files
- Relative files

An unstructured file is an array of bytes of data. The organization of an unstructured file is determined by its creator; unstructured files often contain program code or text.

The other file types are all structured files. A structured file is designed to contain a database. The database contains logical records (individual sets of data about separate items or people). Each type of structured file uses a different structured organization. The *ENSCRIBE Programmer's Guide* describes the three types of structured files.

When you create a file, you can use FUP to specify the structure of the file. You select the file structure to match the type of data you want to store in the file. Section 8, Using FUP for Advanced File Management, demonstrates how to select file characteristics and create both structured and unstructured files with FUP.

In addition to Enscribe files, FUP also supports SQL files. For more information about using FUP with SQL files, see the *File Utility Program (FUP) Reference Manual*.

Disk File Names

A complete (fully qualified) disk file name is unique, and consists of four parts:

Part	Example	Description
Node	\WEST	Physical computer system where the file resides; begins with a backslash (\). All systems must be named. You can omit the node name if the file resides on the current default system.
Volume	$DISK1	Physical disk volume where the file resides; begins with a dollar sign ($), followed by one to seven alphanumeric characters. The character following the dollar sign must be a letter. (Users on remote nodes cannot identify volume names on your node that have more than six characters after the $.)
Subvolume	SUBVOL2	Set of files in the same disk volume; contains from one to either alphanumeric characters and must begin with a letter.
File Identifier	FILE3	Individual file; contains from one to eight alphanumeric characters and must begin with a letter.

Each part is separated by a period, for example:

```
\WEST.$DISK1.SUBVOL2.FILENAME
```

A disk volume is like a file cabinet made up of drawers (subvolumes) containing folders (files). You can name your own subvolumes and files. The node and volume can only be named by the system manager.

A partial (partially qualified) file name is one that omits one or more of the parts.

These are partial file names:

```
FERN.HERST                    SUBVOL2.FILEA
HERST                         FILENAME
```

This is a fully qualified file name for the system \MEL:

```
\MEL.$GERT.FERN.HERST
```

Most disk files are created using application programs. For example, TEDIT creates unstructured EDIT files (file code 101) for your text. You can create structured and unstructured files using the FUP CREATE and TACL CREATE commands. See the *File Utility Program (FUP) Reference Manual* and the *TACL Reference Manual*.

Subvolume Defaulting

When you give a partial file name, the TACL program fills in some of the missing parts using your current default values for node, volume, and subvolume (see Changing Your Default Values on page 3-10).

However, TACL does not supply the subvolume name by default in some situations that were allowed in previous releases. That is, if a volume name is immediately followed by

a file identifier, the TACL program does not recognize it as a valid file name and does not supply the subvolume name. For example, VOL1.MYFILE is not a valid file name, but VOL1.SUBVOL.MYFILE and SUBVOL.MYFILE are valid.

Listing Files and Their Information

The TACL program has several commands that give you information about your files or about other users' files.

Listing Subvolume Contents (FILES Command)

To list the names of all the files in your current default subvolume, enter the FILES command without a subvolume name:

```
1> FILES
$GERT.STEIN

BELL     BOOK     CANDLE    PATIENCE  SARAH
```

To list all the files in any subvolume, enter FILES followed by the subvolume name. For example, to list the files in the subvolume $GERT.FELIX:

```
2> FILES $GERT.FELIX
$GERT.FELIX

AGILITY  CATDOM  CATRWAL   COMIX     FELINES
```

To list the files that reside in a subvolume on another system, give the complete subvolume name:

```
3> FILES \DIVE.$WRECK.RICH
```

Using Wildcards

You can use these wild-card characters to match characters anywhere in a subvolume or volume name (but not a node name):

* Use the asterisk (*) to match zero to eight characters.

? Use the question mark (?) to match a single character.

You cannot use wild cards to match the periods (.) that separate the elements of a file-name string (system, volume, subvolume, and file names). Also, if you use a wild card in the volume name, you must include the dollar sign.

For example, to list all the files in every subvolume on the $SYSTEM disk whose subvolume name begins with the letters SYS:

```
4> FILES $SYSTEM.SYS*
```

You can use more than one wild card in the same command. For example, to list all files that reside in any subvolume that has a three-character name beginning with KH on all volumes beginning with $MT:

```
6> FILES $MT*.KH?
```

Searching For Files With Related Names (FILENAMES Command)

To get a list of all the file names that match a file-name string, use the FILENAMES command and specify the string.

You can use wildcards in your FILENAMES command as described under Using Wildcards on page 3-5.

To search all volumes and all subvolumes for files whose names begin with the word “CLASS”:

```
7> FILENAMES $*.*.CLASS*

$SAG.MEG

CLASSN

$CHANGE.SPUDS

CLASS    CLASSES   CLASSIC
```

It can take some time to check all possible volumes and subvolumes. To stop a FILENAMES command and restore your TACL prompt, press the Break key.

To list every file named HANGUP in every subvolume on every disk volume in the system:

```
8> FILENAMES $*.*.HANGUP
```

To display all five-letter file names in the current subvolume that begin with the letters SEC and end with the number 2:

```
9> FILENAMES SEC?2

$HERST.STEIN

SEC12   SEC22
```

The TACL program displays file names from the current (or default) subvolume because a subvolume name was not included in the command.

Getting File Information (FILEINFO Command)

To list information about a file, include the file name in a FILEINFO command:

```
10> FILEINFO \MEL.$GERT.STEIN.BOOKS
\MEL.$GERT.STEIN
       Code   EOF  Last Modification  Owner   RWEP  PExt  SExt
BOOKS   101 10961  07OCT1998 14:27   8,23  NUUU   18    18
```

For each file, FILEINFO displays the system, volume, subvolume names, and:

Open/Crash indicator	If displayed, indicates the file is open (O) or marked crash-open (?).
Corrupt/SQL DDL/Free Space indicator	If displayed, indicates the file is corrupt (C), and SQL DDL operation is currently in progress (D), or an SQL DDL operation has left unreclaimed free space (F).
Code	File code (EDIT files have file code 101).
Flags (PAL+)	If displayed, indicates a program file with PROGID authority (P), the file is audited (A), licensed (L), or File Format 2(+).
EOF	Number of bytes in the file (also, end of file).
Last Modification	Date and time the file was last modified.
Owner	User ID of the file owner.
RWEP	Read, write, execute, and purge security assigned to the file; `****` is displayed for files protected by Safeguard (security codes are explained in Section 7, Using FUP for Basic File Management), `----` for files that do not exist, and #### for files under OSS security.
PExt SExt	Primary and secondary extent sizes in pages (file extents are explained in Section 8, Using FUP for Advanced File Management).

To display information for all files in your current volume and subvolume, enter FILEINFO without a file name:

```
12> FILEINFO

$GERT.STEIN
         Code   EOF Last Modification   Owner   RWEP  PExt SExt
BELL      101  9872 17MAR1998 06:30:52   8,23   NUUU   12   12
BOOK      101   768 08JUN1998 21:56:29   8,23   OOOO    6    6
CANDLE    101   961 07OCT1998 14:27:23   8,23   NUUU   18   18
PATIENCE  101   566 10JAN1999 07:08:14   8,23   OOOO    4    4
SARAH    101  1456 10JAN1999 08:14:46   8,23   GOGO   18   18
```

You can use wildcards as described under Using Wildcards on page 3-5.

To display information about all files in the current default subvolume whose names begin with the letter B:

```
13> FILEINFO B*

$GERT.STEIN
       Code   EOF Last Modification   Owner   RWEP  PExt  SExt
BELL    101  9872 17MAR1998 06:30:52   8,23   NUUU   12    12
BOOK    101   768 08JUN1998 21:56:29   8,23   OOOO    6     6
```

To display information about any file in subvolume $GERT.ALICE whose name is six characters long, beginning with SECT and ending with the number 2:

```
14> FILEINFO ALICE.SECT?2
$GERT.ALICE

        Code   EOF Last Modification   Owner   RWEP  PExt  SExt
SECT12   101  2456 10JAN1999 08:14:46   8,23   GOGO   18    18
SECT22   101  5617 10JAN1999 15:23:37   8,23   NUUU   10    10
```

Renaming Files

If you have read and purge access to a file on a disk volume (see Setting Your Default File Security on page 3-13), you can specify a new file name, a new subvolume name, or both.

Simply enter RENAME followed by the old file name, a comma, and the new file name. To rename a file by specifying both a new subvolume name and a new file name:

```
15> RENAME BELL, ALICE.DOORBELL
```

This RENAME command renamed or “moved” the file BELL from the current subvolume to the ALICE subvolume.

You cannot rename a file to another disk volume; the RENAME command does not move files from one disk to another (use the FUP DUP command to place a copy of a file on another volume or subvolume).

You can also rename files with the File Utility Program (FUP). See Renaming Files on page 7-15.

Deleting Files

If you have purge access to a file on a disk volume (see Setting Your Default File Security on page 3-13), and the file is not currently open, you can delete it from the system.

Simply enter PURGE followed by the name of the file to be deleted.

The PURGE command has CONFIRM and NOCONFIRM options that let you specify whether you want the TACL program to confirm your purge request before deleting a file. If you do not enter either option, TACL does not display a confirm prompt.

To verify that files have been purged, enter a FILES command.

Purging Files Using Individual File Names

To delete the file named SECT12 from the subvolume BOOKA on volume $GERT:

```
16> PURGE $GERT.BOOKA.SECT12
$GERT.BOOKA.SECT12 Purged
```

To show how to delete SECT12 using the CONFIRM option:

```
16> PURGE /CONFIRM/ $GERT.BOOKA.SECT12
PURGE $GERT.BOOKA.SECT12 (y/[n])? Y
$GERT.BOOKA.SECT12 Purged
```

You can purge more than one file at a time from the same subvolume or from separate subvolumes by entering a list of file names. Separate the file names with spaces or with commas. For example, these commands show two ways of punctuating a list of file names:

```
17> PURGE SECT01 SECT02 SECT03
SECT01  Purged
SECT02  Purged
SECT03  Purged

18> PURGE STEIN.BOOK,RHALL.TITLE
STEIN.BOOK  Purged
RHALL.TITLE  Purged
```

Purging Files Using File-Name Templates

Another way to purge more than one file at a time is by using file-name templates using an asterisk (*) as a wild card. In this example, all files that begin with SECT are purged. Whether or not you specify a confirm option, the TACL program confirms the file-name template before purging all files that match the template.

```
17> PURGE SECT*
PURGE $STEIN.BOOK.* (y/[n])? Y
$STEIN.BOOK.SECT01  Purged
$STEIN.BOOK.SECT02  Purged
$STEIN.BOOK.SECT03  Purged
```

If you specify the CONFIRM option, the TACL program prompts for purge confirmation of each file before deleting it.

```
17> PURGE /CONFIRM/ SECT*
PURGE $STEIN.BOOK.SECT01 (y/[n])? N
PURGE $STEIN.BOOK.SECT02 (y/[n])? Y
$STEIN.BOOK.SECT02 Purged
PURGE $STEIN.BOOK.SECT03 (y/[n])? N
```

You can also delete files with the File Utility Program (FUP). See Deleting Files on page 7-17.

Changing Your Default Values

Each user has a set of current and previous default values. Each set of defaults includes a value for:

- System (initially, the system where you log on)
- Disk volume
- Subvolume
- File security (see Section 16, Managing Users and Security)

Your saved defaults are in effect when you log on and are your starting point in the system. You can change your saved default system, volume, subvolume, and file security with the DEFAULT program.

Your current defaults define your present location or frame of reference in the system, as you have changed them from your saved defaults since you logged on. During a TACL session, you can:

- Move around on the system or network by changing your current system, volume, and subvolume defaults, using the VOLUME and SYSTEM commands
- Check your current location, using the WHO command
- Display your current defaults in your command prompt, using the SETPROMPT command

File-Name Expansion

When you specify a partial file name in a command, the operating system uses your current default values to supply missing parts of the file name. This adding of parts to file names is known as file-name expansion.

For example, if your current default node is \WEST, default volume is $WRLD, and default subvolume is GLOBE, to purge the file \WEST.$WRLD.GLOBE.SOURCE:

```
20> PURGE SOURCE
```

Using file-name expansion, the operating system assumes that the complete file name is \WEST.$WRLD.GLOBE.SOURCE.

However, the TACL program will not supply a default subvolume name when you supply only a volume name and file identifier, as in $VOL1.MYFILE.

Changing Your Current Default System, Volume, or Subvolume (VOLUME Command)

To change the current default subvolume from $GERT.STEIN (on your home system) to the subvolume RHALL on \LONE.$WELL:

```
1> VOLUME \LONE.$WELL.RHALL
```

After you enter this command, your current defaults are system \LONE, volume $WELL, and subvolume RHALL. For example, the TACL program expands the file name SECT12 to \LONE.$WELL.RHALL.SECT12.

The WHO command shows you that the current volume is now different from your saved volume:

```
2> WHO
Home terminal: $STEIN
TACL process: \MEL.$Z103
Primary CPU: 4 (VLX)   Backup CPU: 5 (VLX)
Default Segment File: $GERT.#6539
  Pages allocated: 8  Pages Maximum: 1024
  Bytes Used: 13364 (0%)  Bytes Maximum: 2097152
Current volume:   $WELL.RHALL    Current system: \LONE
Saved volume:     $GERT.STEIN
Userid: 6,66  Username: SUPPORT.STEIN  Security: "NUNU"
```

To change your current disk volume from \LONE.$WELL to \LONE.$SAG:

```
3> VOLUME $SAG
```

To change your current subvolume from \LONE.$SAG.RHALL to \LONE.$SAG.VITA:

```
4> VOLUME VITA
```

To reset all your current defaults (node, volume, and subvolume) to your saved defaults, enter the VOLUME command with no options:

```
5> VOLUME
6> WHO
Home terminal: $STEIN
TACL process: \MEL.$Z103
Primary CPU: 4 (VLX)   Backup CPU: 5 (VLX)
Default Segment File: $GERT.#6539
  Pages allocated: 8  Pages Maximum: 1024
  Bytes Used: 13364 (0%)  Bytes Maximum: 2097152
Current volume:   $GERT.STEIN
Saved volume:     $GERT.STEIN
Userid: 6,66  Username: SUPPORT.STEIN  Security: "NUNU"
```

You'll notice that the current volume changed from \LONE.$SAG.VITA to $GERT.STEIN on your home system.

Changing Your Current Default Node (SYSTEM Command)

Changing your current default node (system) name lets you omit the node name from the name of a file on a remote system.

To set the current default node name to \LONE:

```
7> SYSTEM \LONE
```

After you enter this command, file names you specify are assumed to reside on node \LONE. If a file on a remote system is available only to local users, you must log on to that system to access the file.

Changing the current default node does not log you onto the other system (to log onto a remote system, see Accessing Other Systems on page 2-6). Entering a command to start a process (such as Peruse) when your current default node is not your saved default starts and runs the process on the other system (subject to network security restrictions).

To reset the current default node to your saved default node, enter SYSTEM without specifying a node name.

Changing Your TACL Prompt (SETPROMPT Command)

Use the SETPROMPT command to change the TACL command prompt displayed on your screen. By default, your TACL prompt displays the current command number. The SETPROMPT command tells the TACL program to include other elements, such as the name of your current default subvolume or volume, or both.

To add your current default subvolume to your TACL prompt:

```
8> SETPROMPT SUBVOL
STEIN 9>
```

To display your current volume in your TACL prompt:

```
STEIN 9> SETPROMPT VOLUME
$GERT 10>
```

To display your current volume and subvolume in the prompt:

```
$GERT 10> SETPROMPT BOTH
$GERT STEIN 11>
```

Now, each time you change your current volume or subvolume, the change is reflected in your prompt:

```
$GERT STEIN 11> VOLUME RHALL
$GERT RHALL 12> VOLUME $WELL
$WELL RHALL 13>
```

To display only the command number again:

```
$GERT STEIN 13> SETPROMPT NONE
14>
```

The TACL program does not maintain your prompt setting between sessions; when you log off, the prompt setting is lost.

To have your TACL prompt automatically customized each time you log on, store a SETPROMPT command in a TACLCSTM file as described in Section 5, Defining Function Keys and Writing Macros.

Changing Your Saved Defaults (DEFAULT Program)

Your saved defaults — including your node, volume, subvolme, and file security — are in effect each time you log on.

Setting Your Default Volume and Subvolume

To change your saved default volume and subvolume:

```
14> DEFAULT $WELL.RHALL
THE DEFAULT <sys-vol-svol> HAS BEEN CHANGED TO $WELL.RHALL.
```

To check your saved default settings, use the WHO command:

```
15> WHO
Home terminal: $STEIN
TACL process: \MEL.$Z103
Primary CPU: 4 (VLX)   Backup CPU: 5 (VLX)
Default Segment File: $GERT.#6539
  Pages allocated: 8  Pages Maximum: 1024
  Bytes Used: 13364 (0%)  Bytes Maximum: 2097152
Current volume:   $GERT.STEIN
Saved volume:     $WELL.RHALL
Userid: 6,66  Username: SUPPORT.STEIN  Security: "NUNU"
```

The DEFAULT command changed the saved volume, not the current volume. Your current location has not changed.

However, the next time you enter a VOLUME command with no command options, or the next time you log on, your current volume will be the new saved volume $WELL.RHALL.

Setting Your Default File Security

When you create a file, the system assigns your saved default file security to the file unless you explicitly assign a different security setting. Changing your default security does not alter the security assigned to previously created files.

The security specifier RWEP is a four-character string. Each position in the string sets the security restriction for one of four disk-file operations:

R who can read the file

W who can write to the file

E who can execute the file

P who can purge the file

In each position, you can use one of these characters:

O (owner)	Only the owner of the file on the local system can access the file. The owner is identified by the user ID associated with the file.
U (user)	Only the owner of the file on the local system, or on the network, can access the file.
G (group)	Any member of the owner's group on the local system can access the file.
C (community)	Any member of the owner's group, either on the local system or on the network, can access the file.
A(anyone)	Any user on the local system can access the file.
N(network)	Any user on the local system, or on the network, can access the file.
-	Only the local super ID (user ID 255,255) can perform the designated operation.

You must enclose a security string in quotes. For example, to change the default security string to NUNU:

```
16> DEFAULT, "NUNU"
THE DEFAULT <file-security> HAS BEEN CHANGED TO "NUNU".
```

The new default security takes effect the next time you log on.

In this example, NUNU, specifies that anyone on the local system, or on the network, can read and execute files that have this security string, but only the owner, anywhere on the network, can write to or purge these files.

With the security string AGOG, anyone on the local system can read files that have this security, anyone in the owner's group can write or purge the files, but only the owner can execute the files. Files with this security string are accessible only on the local system, not over the network.

A security string can also include a hyphen (–), which means that only the local super ID (a user logged on with user ID 255,255) can access the file. However, you cannot include this character in your default security string; you can set it only with the FUP SECURE command.

4

Starting and Controlling Processes With TACL

A process is a running program that is started from a program object file on disk. For example, $SYSTEM.SYSTEM.TFORM is a program object file. When you run TFORM, you start a TFORM process.

You can manage the processes on your system by using the TACL program.

Selected TACL commands are described in this section. For information about all TACL commands and programs, see the *TACL Reference Manual*.

Getting Information About Processes

When you encounter problems with a process, you will need to get more information about the processes that are running on your system, and their status.

Use the TACL PPD (process-pair directory) or the TACL STATUS command to get information about a process, learn what process pairs are currently running, and obtain information about them.

Displaying Process Information (STATUS Command)

To display information about one or more running processes, including the process name, the user ID associated with the process, the location of the program file, and the home terminal from which the process was started, use the STATUS command:

```
> STATUS process-id
```

The STATUS command accepts several command options. For example, to display information about all processes running from your home terminal, use an asterisk (*) to indicate all processes, and TERM to indicate the home terminal:

```
1> STATUS *, TERM $term-name
Process       Pri PFR %WT Userid Program file          Hometerm
        3,87  148     005  6,66  $GERT.ALWORK.TEDIT    $STEIN
$Z103   4,140 150   R 000  6,66  $SYSTEM.SYSTEM.TACL    $STEIN
$Z103 B 5,69  150     001  6,66  $SYSTEM.SYSTEM.TACL    $STEIN
```

You can also specify a user and program by name. For example, the following STATUS command shows all processes owned by the user COOKS.SAG that are running the program file $SYSTEM.SYSTEM.TEDIT:

```
2> STATUS *, USER COOKS.SAG, PROG $SYSTEM.SYSTEM.TEDIT
Process       Pri  PFR  WT% Userid  Program file          Hometerm
3,67          148    R  001   7,1   $SYSTEM.SYSTEM.TEDIT  $STEIN
```

For a description of all STATUS command options, see the *TACL Reference Manual*.

Examples

1. Display information about processes running on the terminal $JT1.#J01.

   ```
   > STATUS *, TERM $JT1.#J01
   ```

   A report such as this is sent to your home terminal:

```
Process         Pri PFR %WT Userid  Program file              Hometerm
$JT12  B  4,85  150     001   8,001 $SYSTEM.SYS02.TACL        $JT1.#J01
$JT12     5,84  150   R 000   8,001 $SYSTEM.SYS02.TACL        $JT1.#J01
          9,151 140     004   8,001 $SYSTEM.SYSTEM.SUBVOLS    $JT1.#J012
```

2. Display information about the process 9,151.

   ```
   > STATUS 9,151
   ```

A report such as this is sent to your home terminal:

```
System \SAGE

Process         Pri PFR %WT Userid  Program file               Hometerm
          9,151 140     004   8,001 $SYSTEM.SYSTEM.SUBVOLS     $JT1.#J01
                    Swap File Name: $DATA1.#8890
```

Interpreting STATUS Command Displays

The following describes the elements of the STATUS command display:

```
  1             2   3   4      5            6                      7
Process         Pri PFR %WT Userid  Program file               Hometerm
$JT12  B  4,85  150     001   8,001 $SYSTEM.SYS02.TACL         $JT1.#J01
$JT12     5,84  150   R 000   8,001 $SYSTEM.SYS02.TACL         $JT1.#J01
          9,151 140     004   8,001 $SYSTEM.SYSTEM.SUBVOLS     $JT1.#J012
```

1 `Process` is the name of the process (whether or not it is a backup process), the number of the processor, and the process identification number (PIN). “B” indicates a backup process. A blank indicates an unnamed process.

The second column shows the CPU number where the process is running and the PIN for the process in that CPU.

2 `Pri` is the execution priority of the process.

3 `PFR` can contain the following codes:

- P The process contains privileged code.
- F The process is waiting for a page fault.
- R The process is on the ready list.

4 `%WT` is the process wait state (in octal). A value of 000 indicates the process is not waiting.

5 `Userid` is the group and user ID number under which the process is running.

6 `Program` file is the name of the program file that started the process.

7 `Hometerm` is the name of the terminal that is designated as the home terminal for the process, which is where the process was started.

Displaying Named Process Information (PPD Command)

The PPD command displays the names, process IDs, and ancestors (the process creators) of named processes currently in the destination control table (DCT), a table that contains information about processes.

Use the PPD command if you know the name of a process and want to find out its CPU and process identification number (PIN) (and the CPU and PIN of its backup, if one exists), or if you want to find out what process created a named process.

To display all named processes running on your system:

```
> PPD
```

To display information about a specific named process that is running on your system:

```
> PPD $process-name
```

Note. Because large numbers of processes might be running on your system at any time, the PPD display can be lengthy. If you want to stop the display, press the Break key on your keyboard. For the same reason, you might want to send the PPD display to a disk file or printer. Refer to the first example, below, for instructions on how to perform this task.

PPD lists the name of the process, the CPU and PIN of the primary and backup processes, and the name or CPU and PIN of the ancestor process. If you don't specify a process, PPD displays information about all named processes on the system.

For example, suppose a process called $MGMT is running on the system. To find out who started the process:

```
3> PPD $MGMT
Name   Primary Backup  Ancestor
$MGMT  2,21    3,20    $Z048
```

The Ancestor column indicates that $Z048 is the ancestor (the process that created $MGMT). You can now enter a STATUS command using the name of the ancestor, $Z048:

```
4> STATUS $Z048
Process       Pri  PFR  WT% Userid  Program file          Hometerm
$Z048   0,12  150       005   0,0   $SYSTEM.SYSTEM.TACL   $OPCONS
                    Swap File Name: $TEMP.#0170
$Z048 B 1,14  150       001   0,0   $SYSTEM.SYSTEM.TACL   $OPCONS
                    Swap File Name: $TEMP.#0170
```

$Z048 is the TACL process on the terminal named $OPCONS, which on this system is the operator console. The null user ID (0,0) indicates that no one is logged onto this TACL process.

Examples

1. Send a listing of all running processes to your subvolume CURRENT and your file PROCESS.

   ```
   > PPD / OUT CURRENT.PROCESS /
   ```

A report such as this is sent to the file named PROCESS in the subvolume named CURRENT:

```
Name    Primary Backup  Ancestor
$ZL00    0,3
$Z001    1,51    0,69   $BA1
$ZSCB    1,113   0,70   $ZSCA
$ZLOG    0,47    1,47   $ZTAS
$ZELM    0,51    1,49   $ZTAS
$ZL01    1,3
   .      .       .       .
   .      .       .       .
   .      .       .       .
```

2. View information about the process $WOW.

   ```
   > PPD $WOW
   ```

   A report such as this is sent to your home terminal:

```
Name    Primary Backup  Ancestor
$WOW    04,054  05,009  $Z000
```

Interpreting PPD Command Displays

The elements of the PPD command display and their meanings are:

```
 1       2       3       4
Name    Primary Backup  Ancestor
$WOW    04,054  05,009  $Z000
```

1	`Name`	Name of the process
2	`Primary`	Processor number and process number of the primary process in a process pair, or of the specified process if it is not a member of a pair
3	`Backup`	Processor number and process number of the backup process in a process pair. If no value is displayed, the process has no backup.
4	`Ancestor`	Identity of the process that created the process listed under Name.

If a process you specify is not running, the message “PROCESS DOES NOT EXIST” is displayed.

Starting and Controlling a Process

To start a process from the TACL program, enter the TACL RUN command. A RUN command names the program object file that you want to run. You can also specify command options, such as input and output files to be used by the program, the processor where the process runs, the name and execution priority of the process, and the number of data pages to be used.

You can enter a RUN command either explicitly or implicitly:

- An explicit RUN command includes the keyword RUN followed by the name of the program file:

```
5> RUN $SYSTEM.OPERATE.SWAPURGE
```

  If you do not include the volume and subvolume, the TACL program expands the file name using your current defaults.

- An implicit RUN command includes the name of the program file, without the keyword RUN:

```
6> TEDIT
```

  If you specify a partial file name in an implicit RUN command (as for TEDIT in this example), the TACL program does not expand the file name. Instead, it searches for the program on $SYSTEM.SYSTEM. If the TACL program cannot find the program there, you receive an error. (You can include #PMSEARCHLIST in your TACLCSTM file to tell the TACL program to search in your defaults; see Section 5, Defining Function Keys and Writing Macros.)

Running a Process at a High PIN

Using high process identification numbers (PINs) lets you concurrently run more processes. High PINs have values above 255; low PINs have values from 0 through 254. Because some processes and devices must run at low PINs, you might want to run an application at a high PIN if there is a shortage of low PINs. To run the application ACCOUNTS at a high PIN, enter:

```
7> RUN ACCOUNTS /HIGHPIN ON/
```

HIGHPIN ON specifies that the program will run at a high PIN if the HIGHPIN flag is set in the object file (and library file, if any) and if a high PIN is available. For more information about running a program at a high PIN from the TACL program, see the *TACL Reference Manual*.

Your Default Process

When you start a new process, the TACL program stores the process name, its CPU, and PIN in a special buffer that holds one process name and number at a time. This buffer identifies your default process.

If you enter a process-control command (such as PAUSE, ACTIVATE, STOP, and SUSPEND) but do not specify a process name or CPU and PIN, the TACL program assumes that you are referring to the default process. The default process is cleared when that process terminates.

Interrupting a Process

Use the Break key to interrupt a process and return to the TACL program. When you press Break, most processes yield control of the terminal to the TACL program and continue to run in background mode. If a background process requires input from or

output to the terminal, the process waits indefinitely until it can control the terminal again or until it stops.

While that process is running in the background, your TACL prompt is active, and you can enter TACL commands. For example, you can use the STATUS command to see what processes are still running at your terminal.

Pausing a Process

Use the PAUSE command to let a background process gain control of your terminal. If you press Break while a process is running, or if you include the NOWAIT option when you start a process, you can use PAUSE to pause your TACL process and pass control to the background process.

This example shows the use of the PAUSE command. The dash (-) at the margin is the FUP prompt.

```
8> FUP
File Utility Program - T9074D10 - (08JUN92)    SYSTEM \WEST
Copyright Tandem Computers Incorporated 1981,1983,1985-1992
- (BREAK Key is pressed)
9> STATUS *,TERM

Process       Pri PFR WT% Userid Program file         Hometerm
        2,99  148 P   000   6,18 $SYSTEM.SYS05.FUP    $ALICE
$Z053 B 4,118 150     001   6,18 $SYSTEM.SYSTEM.TACL  $ALICE
$Z053   2,171 150   R 000   6,18 $SYSTEM.SYSTEM.TACL  $ALICE
10> PAUSE
-
```

Breaking out of a process does not stop the process. When you press Break, the TACL prompt (>) appears, but the FUP process continues to run concurrently with the TACL process, as shown by the STATUS command.

When you enter PAUSE, the TACL process is interrupted, and the FUP prompt (-) reappears. After the FUP process stops, the TACL program will redisplay its command prompt (>).

Stopping a Process

Before you shut down your system, you must stop all running applications.

To stop a process that was started incorrectly or that you no longer need, use the STOP command. For example, to stop the process that has CPU and PIN 2,99:

```
13> STOP 2,99
```

To stop the default process (the last process you started), use the STOP command without specifying a process.

Using a Command (OBEY) File

A command file is an EDIT disk file (file code 101) that contains one or more TACL commands. You use the OBEY command to direct the TACL program to read the command file and execute the commands sequentially.

To put comment lines in command files, use the COMMENT command. Comments can occupy more than one line, but each line must begin with the COMMENT command.

This example shows a command file named $GERT.STEIN.NFO that issues the three commands WHO, FILES, and SETPROMPT:

```
COMMENT  --  Execute a WHO Command
WHO
COMMENT  --  List all files in $GERT.STEIN
FILES $GERT.STEIN
COMMENT  --  Set my TACL prompt to my current volume
SETPROMPT VOLUME
```

When you use a command file, the commands and the comments are displayed on the screen. This example invokes the NFO command file shown above. (You can abbreviate the OBEY command with an O.):

```
14> O $GERT.STEIN.NFO
```

The TACL program reads the NFO file, executes the commands sequentially, and displays this information:

```
COMMENT  --  Execute a WHO Command
WHO
Home terminal: $GERT
TACL process: \MEL.$Z103
Primary CPU: 4 (VLX)   Backup CPU: 5 (VLX)
Default Segment File: $GERT.#6539
  Pages allocated: 8  Pages Maximum: 1024
  Bytes Used: 13364 (0%)  Bytes Maximum: 2097152
Current volume:   $GERT.STEIN
Saved volume:     $WELL.RHALL
Userid: 6,66  Username: SUPPORT.STEIN  Security: "NUNU"
COMMENT  --  List all files in $GERT.STEIN
FILES $GERT.STEIN

$GERT.STEIN

BELL     BOOK     CANDLE    NFO     PATIENCE   SARAH
COMMENT Set my TACL prompt to my current volume
SETPROMPT VOLUME
$GERT 15>
```

Restarting a TACL Process

If your TACL process stops, you can restart it by entering a TACL command at another terminal that is still running the TACL program. If you specify the same options for the new TACL process as for the stopped one, you can maintain the original distribution of system resources. To restart the TACL process for a specific terminal you need the:

- Name of the TACL process that controls the terminal (such as $Z103)
- Name of the terminal (such as $STEIN)
- CPU numbers of the processors that run your primary and backup TACL processes (such as 4 and 5)

You can get this information while your TACL process is running by entering the WHO command at your terminal. If your TACL process is not running and you need this information, see your system manager. To restart a TACL process that controls a terminal:

1. Go to another terminal.
2. Enter a command to restart the TACL process, specifying:
   - As both the IN and OUT options, the name of the terminal where you want the TACL process to run
   - As the NAME option, the name of the TACL process
   - As the CPU option, the CPU number of the processor to run the primary process
   - After the final slash (/) following the run options, the CPU number of the processor to run the backup process
   - NOWAIT option

For example, suppose that you have this information for a terminal:

```
15> WHO
Home terminal: $STEIN
TACL process: \MEL.$Z103
Primary CPU: 4 (TXP)   Backup CPU: 5 (TXP)
Default Segment File: $GERT.#6539
Pages allocated: 8  Pages Maximum: 1024
Bytes Used: 13364 (0%)  Bytes Maximum: 2097152
Current volume:   $WELL.RHALL  Current system: \LONE
Saved volume:     $GERT.STEIN
Userid: 6,66  Username: SUPPORT.STEIN  Security: "NUNU"
```

If \MEL.$Z103 is stopped, you can restart it by entering:

```
16> TACL / IN $STEIN, OUT $STEIN, NAME $Z103, CPU 4, NOWAIT/ 5
```

Including the NOWAIT option immediately restores the TACL prompt at the terminal where you enter this command.

For more information about the RUN command and the TACL program, see the *TACL Reference Manual*.

Running Compaq *NonStop*™ Kernel Utilities

Compaq supplies many utility programs. You can start most of these programs from the TACL prompt by entering the program name and pressing Return.

For Peruse, enter:

```
24> PERUSE
PERUSE - T9101D10 - (08JUN92)    SYSTEM  \WEST
Copyright Tandem Computers Incorporated 1978, 1982, 1983, 1984, 1985, 1986,
1987, 1988, 1989
_
   .
   .
   .
_EXIT
```

For the File Utility Program (FUP), enter:

```
25> FUP
File Utility Program - T9074D10 - (08JUN92)    SYSTEM \WEST
Copyright Tandem Computers Incorporated 1981,1983,1985-1992
-
   .
   .
   .
-EXIT
```

Each program displays its startup screen, which includes its program banner, copyright message, and prompt.

Every utility uses a different character for a command prompt. For example, Peruse uses an underscore (_) and FUP uses a hyphen (—).

To exit a utility, enter EXIT at the utility’s prompt.

Solving Common System Process Problems

Table 4-1 lists possible problems, symptoms, causes, and solutions for common process problems.

Table 4-1. Common System Process Problems

Problem	Symptoms	Possible Causes	Solution
A process does not start.	A user gets an error response to a RUN command.	The user might have entered syntax incorrectly.	Reenter the syntax correctly.
		A duplicate process name might exist.	Choose a different name.
		A processor might have failed.	Run the process in a different processor.
		The user might be using a nonexistent program file.	User must use the correct program file.
		Inadequate swap file space might exist.	Make more disk space available.
		A security violation might have occurred.	Obtain security or alter security.
A process stops unexpectedly.	Users cannot access the process.	An application or system error occurs.	Restart the process if you have the proper authority.
	An OMF warning message appears.	A processor failure might have occurred.	Dump the processor's memory and reload it.
		Use INSPECT or DEBUG.	Notify your management.
A process is performing slowly.	Users complain of slow response time or "hung" terminals.	A process is looping.	Notify your management.
		A process is suspended.	Use the TACL ACTIVATE command.

5

Defining Function Keys and Writing Macros

Using the TACL program, you can define function keys and write macros to execute frequently used TACL commands, such as FILES, FILEINFO, STATUS, and LOGOFF, or to run an application program and pass parameters to the application.

Using function keys and macros, you can reduce:

- The length of commands you must type by defining a function key or creating a short macro name for a long command or a series of commands
- Problems caused by typing errors by reducing the typing you must do for common commands
- The amount you must remember or look up the command syntax

For more information on these topics, see the *TACL Reference Manual* and the *TACL Programming Guide*.

Defining and Using Your Function Keys

There are 32 function keys you can define for your use:

- Function keys F1 through F16
- Shifted keys SF1 through SF16 (hold down the shift key while pressing a function key F1 through F16)

F16 is predefined as the TACL HELP key, but you can redefine it using the #HELPKEY built-in variable, as described in the *TACL Reference Manual*.

You can create two types of function-key definitions:

Type	Description	Example
Alias	Defines a function key as a name for a TACL command or application.	Define the F1 key as an alias for the FILES command; when you press F1, the TACL program executes this command.
Macro	Defines a function key to invoke one or more TACL commands or run an application.	To specify arguments for a macro, enter them on the TACL command line before you press the function key.

Creating a Library File for Your Function Keys

To define your function keys, you create a library file that contains the function-key definitions. Your library file is an EDIT file (file code 101). Create this file in your saved default subvolume, using a text editor such as TEDIT. Give the file a descriptive name such as MYKEYS.

For each function key you want to define, enter:

```
?SECTION function-key type
command
[ command ]...
```

`?SECTION`

TACL directive that indicates the beginning of a definition.

`function-key`

Name of the function key you are defining. Unshifted function keys (F1 through F16) or shifted function keys (SF1 through SF16).

`type`

Type of definition: ALIAS or MACRO. Other types of function-key definitions are described in the *TACL Reference Manual.*

`command`

TACL command that you want this function key to execute.

Inserting Comments in a Library File

You can add comments to your library file in three ways:

- Begin a comment line with the COMMENT command.
- Enclose the comment in braces ({ }). You can embed a braced comment in the middle of a command.
- Introduce the comment with two equal signs (==). A comment beginning with == must either begin a line or follow all other text on the line.

TACL ignores comments when you press a defined function key.

Writing an Alias Definition

An alias definition provides an alternate name for a TACL command. For example, the F1 key can be an alias for the FILES command as shown in the example below.

To include comments in an alias definition, use braces ({ }) or double equal signs (==); you cannot use the COMMENT command because this definition can contain only one command.

These examples show alias key definitions for some frequently used commands:

```
?SECTION F1 ALIAS
FILES

?SECTION F2 ALIAS
FILE{Display information about my files}INFO

?SECTION F3 ALIAS
PERUSE
== Start PERUSE

?SECTION F4 ALIAS
FILENAMES
== Display file names

?SECTION F5 ALIAS
== Display the date and time
TIME

?SECTION SF16 ALIAS
LOGOFF
```

After you load these definitions, function keys F1 through F5 will be aliases for the commands FILES, FILEINFO, PERUSE, FILENAMES, and TIME, respectively. SF16 is an alias for the LOGOFF command.

If the command included in an alias definition accepts command arguments, you can enter values for these arguments when you use the function key, as described in Passing a Value to an Alias Definition on page 5-7.

Writing a Macro Definition

You create macro definitions similar to the way you create alias definitions. A macro definition can invoke multiple commands and can include specific command arguments and dummy arguments for which you specify values when you use the defined key.

To create macro definitions, use the MYKEYS file again. Begin each function-key definition with the ?SECTION command, and specify the definition type as MACRO. You can also add comments using the COMMENT command, braces ({ }), and double equal signs (==).

Including Command Arguments

In a macro definition, you can include command arguments using two methods:

- Include the specific argument in the definition.

  For example, the STATUS command can take the argument *, TERM. Therefore, your MYKEYS file could define function key F6:

  ```
  ?SECTION F6 MACRO
  STATUS *, TERM
  COMMENT Info about processes at this terminal
  ```

- Include a dummy argument for each variable.

  You can include a dummy argument for each variable in the command. A dummy argument is a number surrounded by percent signs (such as %1%, %2%, and %3%). Then, before you press the function key, type the actual values for each variable on the TACL command line.

  The dummy arguments in this TFORM command are replaced by the real values that you specify before you press F7:

  ```
  ?SECTION F7 MACRO
  TFORM / IN %1%, OUT $S.#%2%, NOWAIT /
  ```

  This macro definition contains all constant elements in the TFORM command, which are the run-option keywords IN, OUT, and NOWAIT, the first part of the spooler location ($S), and all required punctuation (forward slashes, commas, and the pound sign). The two dummy arguments (%1% and %2%) represent the variables, a file name and spooler location , which you enter when you use F7.

  The dummy argument %*% represents any number of arguments that you supply. You must also include any required punctuation. For example, you can define the STATUS command using the %*% argument:

  ```
  ?SECTION F1 MACRO
  STATUS %*%
  ```

  To enter a STATUS *, TERM command using a key with this definition, enter:

  ```
  15> *, TERM
  ```

  and press F1.

Differences Between Alias and Macro Definitions

Although alias and macro definitions appear similar, they are different. An alias definition is a single command name. It cannot contain command or dummy arguments, although you can pass arguments when you use the definition. A macro definition can contain command arguments and dummy arguments. A macro definition can also contain a sequence of TACL commands.

This example shows three ways to write a function-key definition for the STATUS command:

```
?SECTION F8 ALIAS
STATUS

?SECTION F8 MACRO
STATUS *, TERM
COMMENT Info about processes running at this terminal

?SECTION F8 MACRO
STATUS %1%, %2% %3%
```

- The first example uses an alias definition because it defines the F8 key to invoke the STATUS command. The STATUS command alone uses the default process as an argument. To pass arguments to the STATUS command when you invoke it, enter the values in proper order, including all required punctuation, and then press F8. For example, enter:

```
12> *, USER
```

  Then press F8.

- The second example uses a macro definition because it executes a command that includes two command arguments and all required punctuation. It also contains two commands (STATUS and COMMENT). To execute the STATUS *, TERM command, you simply press F8.

- The third example also uses a macro definition because it includes three dummy arguments, which you can pass, in proper order, any acceptable values. The required punctuation is also included in the definition. To execute the STATUS *, TERM command, you enter:

```
12> * TERM
```

  Then press F8 (no value is passed to the third argument, %3%). This macro definition is further described in Passing a Value to a Macro Definition on page 5-7.

Loading Your Function Key Definitions

To load the library of function-key definitions you created into TACL memory:

```
2> LOAD / KEEP 1 / MYKEYS

Loaded from $GERT.STEIN.MYKEYS:

F1 F2 F3 F4 F5 F6 F7 F8 SF16
```

The KEEP 1 option keeps only the new definitions and deletes any previous ones. If you omit this option, the TACL program keeps all the definitions that you have ever loaded, and you will eventually run out of space.

To load your function-key definitions automatically when you log on, see Customizing Your TACL Environment on page 5-12.

Displaying Your Function Key Definitions

To display a list of your function keys and their definitions, use the KEYS command:

```
13> KEYS
F16  =   (The Help Key)
F1   =   FILES
F2   =   FILEINFO
F3   =   PERUSE
F4   =   FILENAMES
F5   =   TIME
F6   =   STATUS *, TERM
         COMMENT Info about processes
F7   =   TFORM / IN %1%, OUT $S.#%2%, NOWAIT /
F8   =   STATUS %1%, %2% %3%
SF16 =   LOGOFF
```

Comments inserted using the COMMENT command are displayed, but comments inserted with == or { } are not displayed.

Using Your Function Keys

After you load your function-key definitions into TACL memory, you can press a defined function key to execute the associated command.

For example, if you defined F1 as the FILES command, pressing F1 lists the files in your current subvolume:

```
4> $GERT.STEIN

BELL      BOOK      CANDLE    PATIENCE  SARAH
```

If the command defined for a function key accepts command arguments, you can pass a value to the command, and the TACL program uses this value as a command argument. This is true for both alias and macro definitions.

Passing a Value to an Alias Definition

The FILEINFO command (defined as F2) accepts a file name as an argument. By default, when you press F2, you receive information on all files in your current subvolume. To request information for a specific file, enter the name of a file:

```
5> FERN.HERST
```

and press F2. The TACL program displays:

```
$GERT.FERN
         Code   EOF   Last Modification  Owner  RWEP  PExt SExt
HERST    101 12458  02-APR-87 10:55:02  8,56  "NNNN"   12   12
```

Passing a Value to a Macro Definition

To pass values for dummy arguments to a macro definition:

1. Enter the command arguments.
2. Press the function key.

For example, to enter a TFORM command using F7:

1. Enter the name of the file to be printed (this name is substituted for %1%) and the name of the spooler location (substituted for %2%):

```
6> BELL PRINT2
```

2. Press F7.

   This function-key definition is equivalent to:

```
6> TFORM / IN BELL, OUT $S.#PRINT2, NOWAIT /
```

3. Press F3 (the key defined to run PERUSE).

   You can see that your file is spooled to $S.#PRINT2.

Suppose you created this macro function-key definition:

```
?SECTION F8 MACRO
STATUS %1%, %2% %3%
```

In this definition, the comma following the first dummy argument is required by the STATUS command. The first dummy argument represents the range for the STATUS command, the second represents the condition, and the third represents the value for the indicated condition. (For details, see the syntax of the STATUS command in the *TACL Reference Manual*.)

To use this key definition to obtain the status of all processes running in CPU 4 with a priority of 150 or lower:

1. Enter the argument values in order:

```
7> 4 PRI 150
```

2. Press F8.

   The TACL program displays:

```
Process      Pri PFR %WT Userid Program file         Hometerm
$Z43   4,23  150   R 000 8,56   $SYSTEM.SYSTEM.TACL  $STEIN
$Z76 B 4,45  150     001 0,0    $SYSTEM.SYSTEM.TACL  $ABT
$Z55   4,56  150     004 8,24   $SYSTEM.SYSTEM.TACL  $RHALL
```

   The dummy argument %*% accepts any number of command options you might list. For example, you can redefine the F1 key as a macro that uses the dummy argument %*%:

```
?SECTION F1 MACRO
FILES %*%
```

To pass a list of subvolume names to the FILES command:

1. Enter the subvolume names:

```
8> STEIN ABT
```

2. Press F1.

   The TACL program displays:

```
\LONE.$GERT.STEIN
BOOK
\LONE.$GERT.ABT
CHECK
```

Viewing Function Keys in Your Command History

To display the commands you entered and the function keys you pressed at each command line, enter the HISTORY command:

```
9> HISTORY
1> TEDIT MYKEYS
2> LOAD / KEEP 1 / MYKEYS
3> KEYS
4> F1 $GERT.STEIN
5> F2 FERN.HERST
6> F7 BELL PRINT2
7> F8 4 PRI 150
8> F1 STEIN ABT
9> HISTORY
```

For each function key you used, the TACL program displays the name of the key followed by any function arguments you entered before you pressed the key. The TACL Help key (F16) does not appear in the HISTORY display.

Writing TACL Macros

You can write and store TACL macros using:

- A library file, which can contain many macro definitions. Use library files to store macros and aliases for TACL commands that you regularly use.

  To invoke the macro, enter its name at the TACL prompt. The file must be loaded into memory.

- A file that begins with a ?TACL directive, which can contain only one macro definitiony. These files are more appropriate for programmers performing a series of related TACL commands and built-in functions.

  To invoke the macro, enter the name of the file that contains the macro definition. The directive's file is not loaded into memory. For more information about these files, see the *TACL Programming Guide*.

Using a Library File

Creating and loading a general-purpose library file that contains macro definitions is similar to creating and loading a macro that defines your function keys. However, to invoke the macro, you enter the macro name at the TACL prompt instead of pressing a function key.

A library file is an EDIT file (file code 101). Create this file in your saved default subvolume, using a text editor such as TEDIT. Give the file a descriptive name such as MYMACS. For each macro you want to define, enter:

```
?SECTION macro-name macro-type
command
[ command ]...
```

macro-name

Name of the macro you are defining.

macro-type

ALIAS or MACRO. For other types, see the *TACL Reference Manual*.

command

TACL command that you want this macro to execute.

In the first example, the letter P is used as an alias for PERUSE. In the second example, the letters FN invoke the FILENAMES command with one dummy argument.

```
?SECTION P ALIAS
PERUSE

?SECTION FN MACRO
FILENAMES %*%
```

You can add as many alias and macro definitions to this file as you like. You can use dummy arguments, including %*%, in your macro definitions just as you do when defining function keys.

These examples include both alias and macro definitions:

```
?SECTION TED MACRO
TEDIT %*%

?SECTION H MACRO
#OUTPUT Displaying your command history
HISTORY

?SECTION F MACRO
FILEINFO %*%

?SECTION W ALIAS
WHO

?SECTION T MACRO
TFORM / IN %1%, OUT $S.#%2%, NOWAIT /

?SECTION MYINFO MACRO
#OUTPUT Executing the STATUS, WHO, and FILES commands
STATUS *, TERM
WHO
FILES
```

The T macro definition invokes TFORM with two arguments. The first argument is the input text file; the second argument is the spooler location for the output. For example, to run TFORM on a file named INFILE and send the output to the location LAZER:

```
10> T INFILE LAZER
```

The MYINFO macro definition contains a sequence of commands. When you invoke this definition, TACL executes the STATUS, WHO, and FILES commands.

The H and MYINFO macro definitions use the #OUTPUT built-in function to display a message before executing their respective commands.

Loading Your Macro Definitions

After you create your library file of macro definitions, you load the file into TACL memory with the LOAD command or the #LOAD built-in function (like how you load your function-key library file):

```
10> LOAD / KEEP 1 / MYMACS

Loaded from $GERT.STEIN.MYMACS:

P FN ED H F W T INFO
```

Using Your Macros

After you load your macro definitions into TACL memory, you can use your macros.

For example, when you type F, the TACL program executes the FILEINFO command:

```
12> F

$GERT.FERN
        Code   EOF   Last Modification  Owner  RWEP  PExt SExt
HERST    101  12458 02-APR-92 10:55:02  8,56  "NNNN"   12   12
CANDLE   101  16230 07-OCT-92 13:31:36  6,66  "OOOO"    2    2
```

Passing a Value to a Macro

Unlike for function keys, to pass an argument to a macro, enter the macro name and then its arguments.

For example, to pass the specific name of a file to your FILEINFO macro:

```
11> F CANDLE

$GERT.STEIN
       Code    EOF Last Modification  Owner   RWEP  PExt SExt
CANDLE  101 16230 7-OCT-92 13:31:36   6,66  "OOOO"    2    2
```

Using a File Starting With a ?TACL Directive

You can also write a TACL macro by entering the ?TACL MACRO directive and the commands you want the macro to perform into an EDIT file (file code 101).

To create a macro that performs the WHO command.

```
15> TEDIT MYSTAT

?TACL MACRO
WHO
```

To execute the macro, enter:

```
16> RUN MYSTAT
```

Files that contain a ?TACL MACRO directive cannot be loaded into memory, as library files are. You can store only one macro definition in a file that starts with the ?TACL MACRO directive. Instead, use the ?SECTION directive to define macros in files that you plan to load.

Customizing Your TACL Environment

When you log on, the TACL program customizes your environment according to the commands in the TACLLOCL and TACLCSTM files. Your system management creates a single TACLLOCL file for everyone on the system. You create your TACLCSTM file in your saved default subvolume. When you log on, the TACL program automatically executes the commands in these two files.

Instead of using the LOAD command each time you log on to use your own function-key definitions and macros, add the LOAD command for each library file to your TACLCSTM file. Thus, the TACL program automatically loads your function-key and macro definitions whenever you log on.

Note. You cannot load definitions stored in files that begin with the ?TACL MACRO directive.

---

1. Create an EDIT file (file code 101) named TACLCSTM in your saved default subvolume. (On some systems, TACLCSTM file is created for you.)
2. In this file, enter these commands to load your library files:

   ```
   ?TACL MACRO
   COMMENT TACLCSTM file
   LOAD / KEEP 1 / MYKEYS {Load My Key Definitions}
   LOAD / KEEP 1 / MYMACS {Load My Macro Definitions}
   ```

   Your TACLCSTM file must begin with a ?TACL MACRO. The LOAD commands in this example are the same as those you would use to load your library files every time you log on.
3. Add comments as needed, using the COMMENT command, the double equal signs (==), or braces ({}).
4. Add any other commands that you want to execute when you log on.

   For example, you can enter a SETPROMPT command so that your TACL prompt contains the name of your current volume and subvolume. To automatically execute this command whenever you log on, add it to your TACLCSTM file:

   ```
   ?TACL MACRO
   LOAD / KEEP 1 / MYKEYS {Load My Key Definitions}
   LOAD / KEEP 1 / MYMACS {Load My Macro Definitions}
   SETPROMPT BOTH
   ```

   You can add other built-in TACL functions to your TACLCSTM file. For example:

   ```
   == Set the default search list
   #SET #PMSEARCHLIST #DEFAULTS $SYSTEM.SYSTEM
   == Set the TACL input format
   #SET #INFORMAT TACL
   == Set the TACL output format
   #SET #OUTFORMAT TACL
   ```

To load any function-key definitions you changed while logged on, issue the LOAD command to load the new definitions or log off and log on again.

See the *TACL Reference Manual* for more information about the TACLCSTM file.

6 Creating and Using DEFINEs

A DEFINE is a named set of attributes and values that let you specify information for a process before you start the process.

The DEFAULTS DEFINE is a simple example; this DEFINE contains your default node, volume, and subvolume names, which the system uses to expand your partially qualified file names.

There are several advantages to using a DEFINE:

- Convenience. For example, a MAP DEFINE lets you substitute a logical name for an actual file name. It's easier to use a DEFINE name such as =CUSTOMERS than an actual file name such as \NY.$ACCNTS.CURRNT.CUSTNMES.
- Saving time. Setting attributes for a spooler job with a SPOOL DEFINE is faster than setting the attributes with Peruse after the job is on the print queue.
- Using a TAPE DEFINE lets you access labeled tapes (if labeled-tape processing is enabled for your system).

To use a DEFINE, you should be familiar with the TACL program, including its relationship to other processes running on your system. See Section 4, Starting and Controlling Processes With TACL.

Using DEFINEs with FUP, labeled tapes, and Backup and Restore is described in Section 7, Using FUP for Basic File Management, Section 10, Using Labeled Tapes, and Section 11, Backing Up and Restoring Disk Information, respectively.

Using a DEFINE

A DEFINE lets you specify information for a process before you start the process.

Considerations When Using DEFINEs

- The TACL program provides commands with which you create, list, modify, and delete your DEFINEs.
- TACL stores DEFINEs in its process file segment (PFS). When you start a process from TACL, you can specify whether your DEFINEs are propagated to the new process with the DEFMODE attribute setting.
- A DEFINE remains in effect only for the current TACL session. After you log off, the DEFINE no longer exists. (The TACL #DEFINESAVE and #DEFINERESTORE built-in functions let you save a DEFINE during one TACL session and then restore the DEFINE during a subsequent session. See the *TACL Reference Manual* for information about TACL built-in functions.)
- If you start a process and then create a DEFINE for this process, the DEFINE has no effect on the process. You must create the DEFINE before you start the process.
- You can store the commands that create a DEFINE in a command file. When you want to use the DEFINE, you simply execute the command file. You can also edit the command file if you want to change the DEFINE.

DEFINE Names

A DEFINE name is a logical name that:

- Contains from 2 to 24 characters.
- The first character must be an equal sign (=).
- The second character must be a letter.
- The remaining characters can be letters, numbers, hyphens (-), underscores (_), or circumflexes (^).
- When specified as the value of a system procedure parameter that has a fixed length of 24 characters, a DEFINE name must be left-justified in the DEFINE name buffer and padded on the right with blanks.

Uppercase and lowercase letters in a DEFINE name are equivalent.

Note. Compaq reserves all DEFINE names beginning with an equal sign and underscore (=_). Do not use DEFINE names that begin with this two-character string.

Examples of DEFINE names are:

```
=CUSTOMER^RECORDS
=Accounts_File
=TEST_RESULTS_FOR_JANUARY
=TAPE-FILE
=disk_file_records
=Customer^Records
=Y
=N^^^---^^^
```

Uppercase and lowercase letters are equivalent. In the above examples, =CUSTOMER^RECORDS is equivalent to =Customer^Records.

DEFINE Templates

Some DEFINE commands let you use a template in place of an actual DEFINE name. DEFINE templates let you use certain characters to make a command or function operate on a number of similarly named DEFINEs. Characters valid in DEFINE templates are:

*	Matches 0 or more characters at the same position (similar to a file-name template)
?	Matches one character in the same position (similar to a file-name template)
** or =*	Matches all DEFINE names

For example, the DEFINE template =CUST* matches all DEFINE names that begin with =CUST.

DEFINE Classes

The primary attribute of a DEFINE is the CLASS attribute, which identifies the DEFINE subtype.

Class	Specifies
CATALOG	SQL catalog name that substitutes for the DEFINE name in a program
DEFAULTS	Process defaults, such as the default volume and subvolume
MAP	A file name that subsitutes for the DEFINE name in a program
SEARCH	Information to be used for resolving file names with a search list
SORT, SUBSORT	Parameters for the FASTSORT program
SPOOL	Attributes of a spooler job, such as location and number of copies
TAPE	Attributes of a file on a labeled tape, such as block size and density

The CATALOG, DEFAULTS , MAP, SPOOL, and TAPE classes are described below. The SEARCH class is described in the *Guardian Programmer's Guide* and the *Guardian Procedure Calls Reference Manual*; SORT and SUBSORT classes are described in the *TACL Reference Manual* and the *FastSort Manual.*

CATALOG DEFINE

A CATALOG DEFINE lets you specify a logical name for a particular SQL catalog. (The CATALOG DEFINE does not change the current default catalog.)

For example, this CATALOG DEFINE lets you substitute =PCAT for the catalog that resides on subvolume \ACCT.$DISK.CAT:

```
3> ADD DEFINE =PCAT, CLASS CATALOG, SUBVOL \ACCT.$DISK.CAT
4> INFO DEFINE =PCAT
   Define Name         =PCAT
   CLASS               CATALOG
   FILE                \ACCT.$DISK.CAT
```

For more information, see the *NonStop SQL/MP Reference Manual*.

DEFAULTS DEFINE

The DEFAULTS DEFINE contains standard default values such as the default volume and subvolume names. Each process has a DEFAULTS DEFINE with the name =_DEFAULTS. (This DEFINE name contains an underscore (_) as the second character because it is a Compaq DEFINE.)

This example shows the DEFAULTS DEFINE displayed with the TACL INFO DEFINE command:

```
8> INFO DEFINE =_DEFAULTS
   Define Name         =_DEFAULTS
   CLASS               DEFAULTS
   VOLUME              $DATA5.MYSUBVOL
```

The DEFAULTS DEFINE:

- Is always propagated to a new process, regardless of the DEFMODE setting
- Cannot be deleted or renamed, but can be displayed and altered
- Is automatically modified by the TACL program to match its current volume setting and is set from the default volume in the startup message

Attributes for the DEFAULTS DEFINE are VOLUME, SWAP, and CATALOG, which are described in the *TACL Reference Manual.*

MAP DEFINE

A MAP DEFINE lets you substitute a logical DEFINE name for an actual file name. When you create a MAP DEFINE, you give it the name you want to use as a substitute for an actual file name. You can use a MAP DEFINE wherever a file name can be used.

For example, this MAP DEFINE lets you substitute =JUNE^RECORDS for the file name \ACCT.$DISK.RECORDS.JUNE when you enter a command that uses this file name:

```
4> ADD DEFINE =JUNE^RECORDS, CLASS MAP, FILE \ACCT.$DISK.RECORDS.JUNE
5> INFO DEFINE =JUNE^RECORDS
   Define Name         =JUNE^RECORDS
   CLASS               MAP
   FILE                \ACCT.$DISK.RECORDS.JUNE
```

For more information, see the *TACL Reference Manual*.

SPOOL DEFINE

A SPOOL DEFINE lets you set parameters for a spooler job. This DEFINE associates spooler job attributes such as COPIES, REPORT, and LOC with a SPOOL DEFINE name such as =SPOOLER^JOB.

When you start a process that uses the SPOOL DEFINE, the spooler output from that process has the job attributes you specified in the SPOOL DEFINE. You can use a SPOOL DEFINE wherever a spooler collector process name can be used. To display a SPOOL DEFINE with the TACL INFO DEFINE command:

```
6> ADD DEFINE =SPOOLER^JOB, CLASS SPOOL, COPIES 2, LOC $S.#HOLD
7> INFO DEFINE =SPOOLER^JOB
   Define Name         =SPOOLER^JOB
   CLASS               SPOOL
   LOC                 $S.#HOLD
   COPIES              2
```

For more information, see Example of Creating and Using a DEFINE on page 6-9.

TAPE DEFINE

A TAPE DEFINE lets you access a file on a labeled tape. This DEFINE associates labeled-tape attributes such as LABELS, VOLUME, and DENSITY with a DEFINE name such as =TAPE^JOB. You can use a TAPE DEFINE wherever a tape file name can be used.

For example, to transfer a file from a labeled tape to a disk volume, you first create a TAPE DEFINE that contains the attributes of the tape job. To display a TAPE DEFINE with the TACL INFO DEFINE command:

```
5> ADD DEFINE =TAPE^JOB, CLASS TAPE, LABELS ANSI, FILEID INVENTRY
6> INFO DEFINE =TAPE^JOB, DETAIL
   Define Name         =TAPE^JOB
   CLASS               TAPE
   LABELS              ANSI
   FILEID              INVENTRY
```

For more TAPE DEFINE examples, see Section 10, Using Labeled Tapes.

Enabling and Disabling DEFINEs

To enable and disable DEFINEs for any process, including your TACL process, set its DEFMODE attribute. When you set DEFMODE ON or OFF with the TACL SET DEFMODE command, you affect the use of DEFINEs in your current TACL process and in any processes you start:

DEFMODE ON	(Default) DEFINEs are enabled. The system uses existing DEFINEs in your current TACL process and propagates them to any processes you start.
DEFMODE OFF	DEFINEs are disabled. The system ignores any existing DEFINEs in your current TACL process and does not propagate them to processes you start (except the DEFAULTS DEFINE, which is always propagated). You can still alter, display, and delete your DEFINEs, but the system ignores them.

Note. A NonStop™ process pair is an exception. When you create a backup process (one with the same name as the creating process), all DEFINEs are propagated to this process regardless of the current DEFMODE setting. Thus the backup process has the same DEFINEs as the primary process.

To determine the current DEFMODE setting for your TACL process, enter the SHOW DEFMODE command:

```
10> SHOW DEFMODE
        Defmode  OFF
```

If DEFMODE is OFF and you want to use DEFINEs, enter the SET DEFMODE ON command:

```
11> SET DEFMODE ON
```

To prevent a process from using a DEFINE that you have created, enter a SET DEFMODE OFF command:

```
12> SET DEFMODE OFF
```

If you use the TACL RUN command to start a process, you can specify DEFMODE ON or DEFMODE OFF as a run option and this setting overrides your TACL DEFMODE setting. For example, this RUN command starts a process with DEFMODE OFF. Any DEFINEs from your TACL process are not propagated to this new process.

```
13> RUN ACCTPROG / IN $DISK2.FY1989.ACTSFILE, &
13> &OUT $S.LAZR, DEFMODE OFF, NOWAIT /
```

If you start a new process and propagate your DEFINEs to this process, a change made to the original DEFINE in your TACL process does not affect the new process. Conversely, a change in a DEFINE in the new process does not affect your TACL process.

DEFINE Attributes

In addition to the CLASS attribute, each DEFINE has at least one other attribute.

A MAP DEFINE has only the FILE attribute. A CATALOG DEFINE has only a SUBVOL attribute. Attributes for a DEFAULTS DEFINE are VOLUME, SWAP, and CATALOG.

These TAPE DEFINE attributes are described under Using Labeled Tapes With Backup and Restore on page 11-19:

```
BLOCKLEN      FILESECT      OWNER         SYSTEM
DENSITY       FILESEQ       RECFORM       TAPEMODE
DEVICE        GEN           RECLEN        USE
EBCDIC        LABELS        REELS         VERSION
EXPIRATION    MOUNTMSG      RETENTION     VOLUME
FILEID
```

These SPOOL DEFINE attributes are described under Sending Output to a SPOOL DEFINE on page 12-11:

```
BATCHNAME     HOLD          MAXPRINTLINES    REPORT
COPIES        HOLDAFTER     MAXPRINTPAGES    SELPRI
FORM          LOC           OWNER
```

All DEFINE attributes (including SEARCH, SORT, and SUBSORT attributes) are described in detail under the SET DEFINE command in the *TACL Reference Manual*.

Initial Attribute Settings

A DEFINE attribute might have an initial setting, depending on whether it is:

- A default attribute — has an initial default value. For example, the default value for CLASS is MAP.
- A required attribute — does not have an initial value. You must specify a value for each required attribute before you can create a DEFINE. The required attributes for each DEFINE class are:

CATALOG	SUBVOL attribute
DEFAULTS	VOLUME attribute
MAP	FILE attribute
SEARCH	None required
SPOOL	LOC attribute
TAPE	No required attributes, but dependencies exist between attributes (as listed in the *TACL Reference Manual)*. Example: The VOLUME attribute is required if you specify USE IN (if you use the tape file for input).

- An optional attribute has no initial value and does not require a value. OWNER and DENSITY are optional attributes for a TAPE DEFINE.

Working Attribute Set

The working attribute set is a set of DEFINE attributes, that the TACL program maintains.

The working attribute set establishes the attributes and values for the next DEFINE you create, and consists of the attributes and their values associated with the current DEFINE CLASS.

Setting the CLASS attribute establishes a different initial working attribute set for each DEFINE CLASS:

CATALOG	SUBVOL attribute
DEFAULTS	VOLUME, SWAP, and CATALOG attributes
MAP	FILE attribute, which is required but has no default value
SEARCH	SUBVOL attributes, which form a search list that can be used by the FILENAME_RESOLVE_ procedure
SPOOL	All spooler job attributes
TAPE	All tape attributes, with only the LABELS attribute having a value (which is OMITTED)

Each time you set the CLASS attribute, you establish a new working attribute set. Therefore, always set the CLASS attribute first, then set values for the other attributes. (You can set the CLASS attribute when you set other attributes using the ADD DEFINE command; see the SPOOL DEFINE example under Example of Creating and Using a DEFINE on page 6-9.)

For example, when you log on, the CLASS attribute has the default value MAP, and the working attribute set consists of CLASS MAP and the required FILE attribute without an initial value. If you set the CLASS attribute to TAPE, the working attribute set becomes all the tape attributes, but only the LABELS attribute has a value (OMITTED).

Attribute Consistency Checks

When you enter an ADD DEFINE, ALTER DEFINE, or SHOW DEFINE command, the system performs an attribute consistency check on the DEFINE you specified. The system verifies that values are set for all required attributes and performs other consistency checks depending on the DEFINE CLASS. For example, one consistency check ensures that if you specify LABELS ANSI in a TAPE DEFINE, the EBCDIC attribute must be OFF or not specified.

The attribute consistency checks that apply to TAPE DEFINEs are listed in the *TACL Reference Manual*. If a consistency check fails, a message identifies the check number; only one inconsistency is reported for each check.

TACL DEFINE Commands

You can use the TACL DEFINE commands shown in Table 6-1 when working with DEFINEs. For command syntax and options, see the *TACL Reference Manual*.

Table 6-1. TACL DEFINE Commands

Command	Function
SET DEFINE	Sets the values for one or more DEFINE attributes in the working attribute set.
SHOW DEFINE	Displays a value associated with a specific DEFINE attribute, all attribute values that are currently set or defaulted, or all attributes in the working attribute set.
ADD DEFINE	Creates one or more DEFINEs in the process file segment (PFS) of the current TACL process.
RESET DEFINE	Resets one or more attributes in the working attribute set to their initial default values.
INFO DEFINE	Displays the attributes and values associated with one or more existing DEFINEs in the PFS of the current TACL process.
ALTER DEFINE	Changes the attribute values of one or more existing DEFINEs in the PFS of the current TACL process.
DELETE DEFINE	Deletes one or more existing DEFINEs from the PFS of the current TACL process.

Example of Creating and Using a DEFINE

This example shows how to create and use a SPOOL DEFINE with FUP to copy a file from disk to the spooler. The SPOOL DEFINE sets the attributes of the spooler job before you run FUP.

Task 1: Ensure DEFINEs are Enabled

Ensure that DEFINEs are enabled for your TACL process (DEFMODE setting is ON):

```
10> SHOW DEFMODE
        Defmode  ON
```

If DEFMODE is set to OFF, enter a SET DEFMODE ON command:

```
11> SET DEFMODE ON
```

For more information, see Enabling and Disabling DEFINEs on page 6-6.

Task 2: Create the DEFINE

In the ADD DEFINE command, you specify a name and class for the DEFINE, as well as any other attributes you want to define.

1. Enter an ADD DEFINE command to create a SPOOL DEFINE named =FUP-COPY-DEFINE and to set its attributes:

```
12> ADD DEFINE =FUP-COPY-DEFINE, CLASS SPOOL, &
12> &COPIES 2, FORM EMPLST, LOC $S.#LASER, &
12> &HOLDAFTER ON, SELPRI 7, REPORT "Employee Names"
```

   This SPOOL DEFINE specifies that two copies of the job are to be printed on the laser printer associated with location $S.#LASER. The hold-after-printing flag is set so the job will remain in the print queue in case additional copies are required.

   The priority is set to 7 (highest), and the report name is "Employee Names". This DEFINE also sets the FORM attribute to ensure that the job prints on special paper.

2. Enter an INFO DEFINE command for =FUP-COPY-DEFINE with the DETAIL option. Review the displayed attributes and their values.

```
13> INFO DEFINE =FUP-COPY-DEFINE, DETAIL
   Define Name         =FUP-COPY-DEFINE
    CLASS               SPOOL
    LOC                 $S.#LASER
    COPIES              2
    FORM                EMPLST
    HOLDAFTER           ON
    SELPRI              7
    REPORT              EMPLOYEE NAMES
```

Task 3: Use the Created DEFINE

Once a DEFINE is created, use it by refering to it in an appropriate TACL command.

1. Enter a FUP COPY command with the destination file set to the SPOOL DEFINE you just created:

```
14> FUP / OUT FUPMSGS, NOWAIT / &
14> &COPY $DISK2.EMPLOYEE.ALLNAMES, =FUP-COPY-DEFINE
```

   This command also sends any FUP messages to the OUT file FUPMSGS and includes the NOWAIT option so FUP runs in the background.

2. Use Peruse to see the spooler job attributes for the spooler job created by FUP. This job has the attributes specified in your SPOOL DEFINE.

```
15> PERUSE
PERUSE - T9101D10 - (08JUN92)    SYSTEM  \WEST
Copyright Tandem Computers Incorporated 1978, 1982, 1983, 1984, 1985, 1986,
1987, 1988, 1989, 1990, 1991

  JOB  BATCH STATE PAGES COPIES PRI HOLD LOCATION  REPORT
  380        READY 456   2      7    A   #LAZER    EMPLOYEE NAMES
_
```

   After two copies are printed, the job remains on the print queue. You can print additional copies, if needed, or you can delete the job from the queue with the Peruse DEL command.

Part II. Managing Files Using the File Utility Program (FUP)

This part of the guide contains information about using FUP to create and manage files on your Compaq *NonStop*™ Kernel system:

- Section 7, Using FUP for Basic File Management
- Section 8, Using FUP for Advanced File Management

Part II. Managing Files Using the File Utility Program (FUP)

7

Using FUP for Basic File Management

This section describes using the File Utility Program (FUP) for basic disk file management:

For more information about FUP, including the command syntax, see the *File Utility Program (FUP) Reference Manual*.

Who Uses FUP?

Anyone who can log on to the TACL program can use FUP. Although most FUP commands are available to all users, the system enforces security requirements for accessing or altering files. For example, you cannot PURGE a file with FUP unless you have purge access to that file, and you cannot LICENSE a program file that contains privileged code unless you are logged on as the super ID (user ID 255,255). For more information about security, see Section 16, Managing Users and Security.

You can enter FUP commands at the TACL prompt, interactively through FUP itself, or through a command file. You can also programmatically execute some commands from an application. These commands use Subsystem Programmatic Interface (SPI) buffers and procedures for the FUP commands and responses. See the *File Utility Program (FUP) Management Programming Manual* for details.

Before you use FUP to create or manage disk files, you should be familiar with the Compaq *NonStop*™ Kernel operating system concept of a file for the Guardian operating system as described in Section 3, Managing Files With TACL.

Entering FUP Commands

You can enter FUP commands in any of these ways:

- Enter complete FUP commands at the TACL prompt.
- Start a FUP process and then enter commands interactively at the FUP prompt.
- Start a FUP process that reads input from a command file containing FUP commands.

Entering FUP Commands Through TACL

You can enter single FUP commands at the TACL prompt. Enter the term FUP followed by the command. After FUP executes the command, control of the terminal returns to the TACL program.

For example, to enter the FUP INFO * command, type:

```
1> FUP INFO
```

FUP lists file information for each file in your current default subvolume. Then, the TACL prompt reappears.

When you enter FUP commands through the TACL program, you can enter only one command at a time. A separate FUP process starts and completes for each command you enter.

Entering FUP Commands Interactively Through FUP

After you start a FUP process, you can enter commands at the FUP prompt. To start an interactive FUP process, enter FUP with no command name or other options at the TACL prompt. FUP displays its program banner and prompt, a hyphen (-):

```
1> FUP
File Utility Program - T9074D10 - (08JUN92)  SYSTEM \WEST
Copyright Tandem Computers Incorporated 1981,1983,1985-1992
-
```

The FUP process now controls the terminal. The hyphen indicates that the FUP process is ready to accept a command. You can enter a FUP command interactively after the hyphen:

```
-INFO
```

To exit from FUP and return to the TACL program, enter the EXIT command:

```
-EXIT
```

You can also stop a FUP process by pressing the CTRL-Y keys. CTRL-Y marks the end of a file (EOF). When FUP reads CTRL-Y from its input file (in this case, your terminal), it stops execution.

All the examples in this guide, except those in the next subsection, use the interactive method of entering commands through FUP itself.

Entering FUP Commands From a Command File

A command file is an unstructured disk file that contains one or more commands for a specific program. For example, you might create a command file that contains a series of FUP commands. To create a command file, use a text editor, such as TEDIT.

When you start a FUP process, you can include any run option available with the RUN command (described in Section 4, Starting and Controlling Processes With TACL). For example, two common run options are IN (for specifying an input file) and OUT (for an output file).

Sending Input to FUP From a Command File

To execute FUP commands in a command file, specify the command file as the input file with the IN option when you start a FUP process.

For example, these commands, which are in a command file named ALLSUBS, list the subvolumes in the disk volumes $DISK1, $DISK2, and $DISK3:

```
SUBVOLS $DISK1
SUBVOLS $DISK2
SUBVOLS $DISK3
```

To execute these commands, enter a FUP command specifying ALLSUBS as the input file:

```
1> FUP / IN ALLSUBS /
```

Control of the terminal returns to the TACL program after FUP executes the last command in the command file.

You can add comment lines in a command file to identify the file and to explain the operations being performed. FUP comment lines must begin with two dashes (--). Any characters on the line following the -- are ignored by FUP. For example, here is the ALLSUBS command file with comment lines added:

```
-- FUP Commands for Obtaining a List of
-- All Subvolumes in $DISK1, $DISK2, and $DISK3
--
-- Last Modified 1/17/99 13:24
--
SUBVOLS $DISK1    -- Contains manufacturing files
SUBVOLS $DISK2    -- Contains administrative files
SUBVOLS $DISK3    -- Contains all other files
```

Sending Output From FUP to a File

When you start a FUP process, you can use the OUT option to send the process output to a file. The output file (or list file) can be either a disk file or a peripheral device such as a printer. If you do not specify an output file, FUP sends its output to your terminal.

Suppose that you want to save the output from a FUP operation in a disk file. If the output file does not already exist, FUP automatically creates it. If the output file already exists, FUP appends data to it.

In your command to start a FUP process, specify the output file with the OUT option. For example, if ALLSUBS is a command file, and SUBINFO is a file you are using for the output, you can enter:

```
2> FUP / IN ALLSUBS, OUT SUBINFO /
```

To send output to a printer, specify the name of the printer with the OUT option in your FUP command:

```
3> FUP / IN ALLSUBS, OUT $LP /
```

You can also include an output file with individual FUP commands. The OUT file you specify in an individual FUP command overrides any OUT file you specified when you invoked FUP.

For example, these commands create two output files (INFO1 and INFO2) in the current default subvolume, and then send the results of three SUBVOLS commands to those files:

```
4> FUP
-CREATE INFO1
-CREATE INFO2
-SUBVOLS / OUT INFO1 / $DISK1
-SUBVOLS / OUT INFO2 / $DISK2
```

After these three commands are executed, the names of the subvolumes in $DISK1 are listed in the file INFO1, and the names of the subvolumes in $DISK2 are listed in INFO2.

Using DEFINEs in FUP Commands

You can use a SPOOL DEFINE or a TAPE DEFINE in some FUP commands. Use a SPOOL DEFINE to send command output to the spooler. Use a TAPE DEFINE to send command output to a tape file or receive a tape file as input. See Section 6, Creating and Using DEFINEs, for information about creating and using DEFINEs.

Using a SPOOL DEFINE

You can specify a SPOOL DEFINE in these situations:

- As a list file with the OUT option in your command to run FUP:

```
10> FUP / IN $MTERM, OUT =MYSPOOLER /
```

- As a list file with the OUT option in the FILES, HELP, INFO, LISTOPENS, SHOW, and SUBVOLS commands:

```
11> FUP INFO / OUT =DIRECTORY / *
```

- As the destination file in a COPY or BUILDKEYRECORDS command:

```
12> FUP COPY $DATA.RECV.BASE, =MYPRINTER
```

When you use a SPOOL DEFINE with a FUP command, output from the command is spooled to the spooler location that is specified with the LOC attribute in the DEFINE. If this location is a printer, the FUP command output is queued and then printed. Otherwise, the output remains in the spooler, where you can view, redirect, or delete it with Peruse.

This example shows the creation of a SPOOL DEFINE named =MYSPOOLER:

```
1> SET DEFINE CLASS SPOOL, LOC $X5.#LASER, SELPRI 6
2> ADD DEFINE =MYSPOOLER
3> INFO DEFINE =MYSPOOLER, DETAIL
Define name         =MYSPOOLER
CLASS               SPOOL
LOC                 $X5.#LASER
SELPRI              6
```

If you direct FUP command output to =MYSPOOLER, the output is spooled at the destination $X5.#LASER, and the spooler job has priority 6.

Using a TAPE DEFINE

You can use a TAPE DEFINE in place of a tape device name in these situations:

- As the destination in a BUILDKEYRECORDS command
- As either the source or destination in a COPY command
- As the source file in a LOAD command

The syntax of the BUILDKEYRECORDS, COPY, and LOAD commands is shown in the *File Utility Program (FUP) Reference Manual*. Also see the description of the FUP COPY command in that manual for a description of how FUP command parameters interact with TAPE DEFINE attributes.

If you specify LABELS ANSI or LABELS IBM in your DEFINE, your FUP command parameters must not conflict with the DEFINE attributes. For example, if you specify a RECIN value in a COPY command that sends output to a TAPE DEFINE that contains a RECLEN value, the two record-length values must agree or FUP returns an error.

These examples demonstrate the use of TAPE DEFINEs with the FUP COPY command to transfer data from disk to tape, or vice versa.

Because no block or record length is set in the COPY command or in the DEFINE for the first example, FUP uses the default record and block lengths set for the system:

```
1> ADD DEFINE =OUTFILE, CLASS TAPE, LABELS ANSI, VOLUME 88MNOP, USE OUT
2> FUP COPY $DATA.RECDS , =OUTFILE
```

In the next example, the RECLEN attribute of the TAPE DEFINE sets the input record length for the COPY operation to 200 bytes:

```
1> ADD DEFINE =INFILE, CLASS TAPE, LABELS ANSI,VOLUME 89RST, USE OUT,
RECLEN 200
2> FUP COPY =INFILE, $DATA.RECDS
```

Getting Help From FUP

To display a list of the FUP commands, or to display the syntax of a particular FUP command, enter the FUP HELP command. Enter HELP ALL to list all the FUP commands. This example demonstrates how to enter the command and shows the information that you receive:

```
-HELP ALL
ALLOCATE        ALLOW           ALTER           BUILDKEYRECORDS CHECKSUM
COPY            CREATE          DEALLOCATE      DISPLAYBITS     DUP
EXIT            FC              FILES           GIVE            HELP
HISTORY         INFO            LICENSE         LISTLOCKS       LISTOPENS
LOAD            LOADALTFILE     PURGE           PURGEDATA       RELOAD
RENAME          REPORTWIDTH     RESET           RESTART         REVOKE
SECURE          SET             SHOW            STATUS          SUBVOLS
SUSPEND         SYSTEM          VOLUME          !               ?
```

To display the syntax for a specific FUP command, enter HELP followed by the name of the command:

```
-HELP GIVE
GIVE <fileset list> , {  <group id>,<user id>  } [ , PARTONLY ]
                      {<group name>.<user name>}
```

Using the Break Key

When a FUP process is running at your terminal, you can use the Break key as follows:

- If FUP is waiting for a command when you press Break, the TACL prompt reappears, but the FUP process continues to execute in the background.
- If FUP is executing a command that provides information (such as the COPY, FILES, INFO, LISTOPENS, SHOW, and SUBVOLS commands), FUP redisplays its prompt and waits for a new command.
- If FUP is executing any other command, it returns control of the terminal to the TACL program and continues to execute the command in the background.

After the TACL program takes control of your terminal, you can return control to FUP by entering the TACL PAUSE command:

```
20> PAUSE
```

To stop a background FUP process, enter the TACL STOP command. For example, if the last process you started is FUP, which is running in the background, you can stop it by entering:

```
2> STOP
```

Changing System and Volume Defaults

Each FUP process maintains default values for volume and subvolume, as well as a default system in nodes that are part of a network. The default system, volume, and subvolumes for FUP are separate from the default values kept by the TACL program.

FUP expands file names by substituting the current default value for any part of a file name that you omit. Thus, when you enter a file name as part of a FUP command, you need to include only those parts of the name that are different from the current default values kept by FUP. For example, if you supply only a volume name followed by a file identifier, FUP fills in the subvolume name.

Note. You should avoid using subvolume defaulting, because this feature might not be supported in future D-series releases of FUP.

Initially, the default values for a FUP process are the same as the current defaults in effect for your TACL process when you start the FUP process. To change the default values kept by FUP, enter FUP SYSTEM and VOLUME commands.

For example, to change the current default system, enter SYSTEM followed by the name of the system:

```
-SYSTEM \VENICE
```

To return to the initial default system, enter SYSTEM without a system name:

```
-SYSTEM
```

You can change the default volume, the default subvolume, or both with the FUP VOLUME command. To change only the default volume, enter VOLUME followed by the name of the new default volume:

```
-VOLUME $DISK99
```

To change the default subvolume, enter VOLUME followed by the name of the new default subvolume:

```
-VOLUME MAYFLY
```

To change both the default volume and subvolume at once, enter VOLUME followed by the volume name, a period, and the subvolume name. For example, the changes made by the commands in the last two examples can be made at one time by entering:

```
-VOLUME $DISK99.MAYFLY
```

To restore the initial default system, volume, and subvolume values, enter VOLUME alone:

```
-VOLUME
```

Getting Information About Subvolumes and Files

FUP commands can give you the following information:

- A list of the subvolumes in a disk volume (SUBVOLS command)
- A list of the files in a subvolume (FILES command)
- Information about an individual file or file set (INFO command)

To get a list of the subvolumes in a given disk volume, enter the FUP SUBVOLS command followed by the name of the volume:

```
-SUBVOLS $DISK78
```

In systems that are part of a network, a SUBVOLS command can specify a volume in another system:

```
-SUBVOLS \DETROIT.$DISK45
```

If you enter the SUBVOLS command alone, you receive a list of the subvolumes in your current default system and volume:

```
-SUBVOLS
$DISK33
JANIS    JIMIT     MAYA     MICK     NATASHA     SELENA
```

Use the FILES command to get a list of the files in a subvolume. Enter the FILES command followed by the name of the subvolume:

```
-FILES \NY.$APPLE.JACK
```

In the FUP FILES command you must include the system name, volume name, or subvolume name only if it differs from the current default value. For example, this command lists the files in the subvolume SUBTOO in the current default system and volume:

```
-FILES SUBTOO
```

To get a list of the files in every subvolume on a disk volume, enter the FILES command and an asterisk (*) as the subvolume name. For example, this command lists all the subvolumes on the $SYSTEM disk:

```
-FILES $SYSTEM.*
```

Getting Information About Single Files

You can use the FUP INFO command to find out many characteristics of a file or a set of files. Some common uses of the INFO command are described in this subsection. For more information, see Section 8, Using FUP for Advanced File Management, and the description of the FUP INFO command in the *File Utility Program (FUP) Reference Manual*.

To get information about a single file, enter the INFO command followed by the name of the file. This example gets information about the file DICTALT in the current default system, volume, and subvolume:

```
-INFO DICTALT
                 CODE     EOF    LAST MODIF     OWNER   RWEP   TYPE   REC BLOCK
$JUMBO.PATHWAY
DICTALT           101     848   17AUG92 12:22   8,44    NUNU   K      38    1024
```

The first line in the display consists of column headings that identify the information listed. The second line of the display gives the name of the subvolume ($JUMBO.PATHWAY) that contains the listed file (DICTALT).

The headers in the FUP INFO listing indicate:

CODE — the file code. Codes 100 through 999 are reserved by Compaq for system codes. For example, code 100 is assigned to all TNS program object code files, and code 101 is assigned to EDIT files.

EOF — the current length of the file in bytes.

LAST MODIF — the date and time when the file was last modified. If the file was last modified on the day you enter the INFO command, only the time is displayed.

OWNER — the user ID of the owner of the file. Each file is owned by only one user on the system. When a file is created, it is owned by the user who created it.

RWEP — Read/Write/Execute/Purge. This column shows the current security settings for the file; file security is set by the file owner.

For files protected by Safeguard software, this column displays quoted asterisks (****). Use the Safecom program to get information about security for files protected by Safeguard software. For more information about the Safeguard subsystem, see the *Safeguard User's Guide*.

TYPE, REC, and BLOCK show information about structured files. For details, see the description of the FUP INFO command in the *File Utility Program (FUP) Reference Manual*.

Getting Information About File Sets

You can get information about more than one file with a single FUP INFO command. Like several other FUP commands, INFO lets you specify:

- A file set (a set of one or more files)
- A file-set list (a list that includes one or more file sets)

The specification for a file set is much like a single file name, except that a file set can contain more than one file. A file set can be a list of file names separated by commas and enclosed in parentheses; a file set can also be a single file name.

For a file set, you can specify the name of the system, volume, and subvolume where each file resides, just as you would for a single file. If you omit any of these items, FUP expands the file name or names using your current default values.

Wild-Card Characters in File Sets

You can include an asterisk (*) or question mark (?) in place of a volume name, subvolume name, or file identifier.

* (asterisk)	Matches zero to eight characters
? (question mark)	Matches one character

For example, you can get information about all the files in the current default subvolume by entering:

```
-INFO *
```

To get information about all the files in any volume that contains five characters and starts with MANU, enter:

```
-INFO $MANU?.*.*
```

To get information about all the files in the subvolume $MYVOL that start with FILE followed by any two characters, enter:

```
-INFO $MANUF.MYVOL.FILE??
```

File-Set Lists

A file-set list can be a single file set, or a list that includes more than one file set. To include more than one file set in a file-set list:

- Enclose the file-set list in parentheses.
- Include a comma after each file set except the last.

For example, to get information about the files in both the current default subvolume and in the volume $MANUF:

```
-INFO (*, $MANUF.*.*)
```

You can also use the INFO command to get information about files owned by a particular user on the system. Enter the INFO command followed by a file set or a file-set list, a comma, and a user name or user ID, as in these examples:

```
-INFO (*.*, $SYSTEM.PROG1.*), USER MANUF.MABEL
-INFO *, USER 8,44
```

FUP then displays information about only those files in each file-set list that are owned by the specified user. For a complete description of file set and file-set list, see the FUP command syntax summary in the *File Utility Program (FUP) Reference Manual*.

Performing Common File Operations

This subsection describes how to perform basic file management tasks: duplicating, renaming, changing security, and deleting files.

Duplicating Files

You can duplicate a single file or a set of files with the FUP DUPLICATE command. To duplicate a single file, enter DUPLICATE (or simply DUP) followed by the name of the file to be copied, a comma, and the name of the new file.

For example, this command duplicates the file BAKE in your current default subvolume, names the new file BAKE also, and places the copy in the subvolume $PISMO.CLAM:

```
-DUP BAKE, $PISMO.CLAM.BAKE
```

If the file $PISMO.CLAM.BAKE already exists, FUP does not execute the command and responds with an error message. To overwrite an existing file with DUP, you must include the PURGE option:

```
-DUP BAKE, $PISMO.CLAM.BAKE, PURGE
```

To duplicate more than one file with a single command, enter DUP followed by a file set or file-set list, a comma, and a destination.

To specify a destination:

1. Include a system or volume name if either one differs from the current default value.
2. For the subvolume name, specify either of:
   - A single subvolume name, for all the new files to reside in that subvolume.
   - An asterisk (*) in place of the subvolume name (to specify that the subvolume name of each new file be the same as the subvolume name of the file from which it was copied).
3. For the file name, specify either of these:
   - The new file names in order.
   - An asterisk (*) in place of one or more file names if you want the names of the new files to be the same as the old files.

For example, if you have these files:

```
-FILES $ALPHA.SOUP
$ALPHA.SOUP
A       B
-FILES $COUNT.DOWN
$COUNT.DOWN
BLASTOFF
```

- To duplicate the files in both $ALPHA.SOUP and $COUNT.DOWN to the subvolume $SUM.UP, enter:

```
-DUP ($ALPHA.SOUP.*, $COUNT.DOWN.*), $SUM.UP.*
```

Now subvolume $SUM.UP contains copies of all the original files:

```
-FILES $SUM.UP
$SUM.UP
A         B         BLASTOFF
```

- To duplicate the same files to volume $DUKE and retain the original subvolume names, enter:

```
-DUP ($ALPHA.SOUP.*, $COUNT.DOWN.*), $DUKE.*.*
```

Now volume $DUKE has a SOUP and a DOWN subvolume, each containing the files copied from $ALPHA.SOUP and $COUNT.DOWN:

```
-FILES $DUKE.SOUP
$DUKE.SOUP
A         B
-FILES $DUKE.DOWN
$DUKE.DOWN
BLASTOFF
```

If files previously existed in subvolumes $DUKE.SOUP or $DUKE.DOWN, they would also be listed by this FILES command.

- You can also use * as a wild-card character. For example, to duplicate all files on OLDSVOL that end in FILE, enter:

```
-DUP $ALPHA.OLDSVOL.*FILE, $ALPHA.NEWSVOL.*
```

Duplicating a File Using the RESTARTABLE Option

When duplicating large files, especially over a network that might experience problems, use the RESTARTABLE option. That way, in case the duplicate operation fails before it is finished, a subsequent RESTART command continues the duplicate operation from the point of failure rather than from the beginning, saving system time and resources.

FUP saves the restart information in a disk restart file (file code 855). You can name this restart file in your DUPLICATE command, or you can let FUP create the ZZRSTART file in your default subvolume. The RESTART command uses the information in this file to restart the duplicate operation from the point of failure.

1. Duplicate the file BIGFILE on system \WEST to NEWFILE on system \EAST and use the RESTARTABLE option with RSFILE as the restart file, enter:

```
-DUP \WEST.$DISK2.ACCTS.BIGFILE, &
-\EAST.$DISK4.ACCTS.NEWFILE, RESTARTABLE RSFILE
```

If you had not supplied the RSFILE file name, FUP would have created the ZZRSTART file in your default subvolume as the restart file.

2. If the duplicate operation fails before it is finished, restart the operation:

```
-RESTART RSFILE
```

FUP displays the original DUPLICATE command and continues the operation from the point of failure. FUP continues to update the restart file. If a second failure occurs, a RESTART command restarts the operation from the second point of failure.

For more information, see the *File Utility Program (FUP) Reference Manual*.

Renaming Files

Use the FUP RENAME command to rename a file or a set of files. To rename a single file, enter RENAME followed by the current file name, a comma, and the new file name:

```
-RENAME FRED.INFO, MABEL.ARCHIVE
```

In this example, MABEL.ARCHIVE is the new name of the file. Note that you can change the subvolume name and the file name, but not the volume name; a file that is renamed remains on the same disk volume. To duplicate a file to another disk volume, use the FUP DUPLICATE command.

To rename a set of files, specify a file set just as you would for the DUPLICATE command. When you RENAME a file set, however, all the file names must remain the same; only the subvolume name can change. Enter RENAME followed by a file set or file-set list to be renamed, a comma, and a destination. For the destination, you must specify:

- A subvolume name different from the subvolume names in the file set or file-set list to be renamed; the renamed files will reside in this different subvolume.
- An asterisk (*) in place of the file name; the names of the renamed files will be the same as those of the original files.

Suppose you want to rename the files in subvolumes $BIG.DOCUMNTS and $BIG.PROGDOCS. The renamed files must still reside in volume $BIG, but you would like to put all of them in the subvolume ALLDOCS. Enter:

```
-RENAME ($BIG.DOCUMNTS.*, $BIG.PROGDOCS.*), $BIG.ALLDOCS.*
```

If you omit system names or volume names in the file set or file-set list, FUP assumes the current default values. Also, you cannot change the volume names of files with the RENAME command. If you include a system or volume name in the file set or file-set list to be renamed, the same system or volume name must appear in the destination.

Changing File Security

If you own a file, you can use the FUP SECURE command to change its security by assigning the file a new security string — four characters that specify who can read, write, execute, and purge a file.

Your logon default security string is automatically assigned to files you create, though you can specify a different security for specific files. You can also change your default security string as described in Section 3, Managing Files With TACL. Table 7-1 lists the characters you can use in a security string. “Local” refers to access within a single system; “remote” refers to access between systems (nodes) in a network.

Table 7-1. Levels of File Security

FUP Code	Program Value	Who Can Access the File
–	7	Local super ID only
U	6	Local or remote owner (any user with the owner’s user ID)
C	5	Local or remote member of the owner’s group
N	4	Any local or remote user
O	2	Local owner only
G	1	Local member of the owner’s group
A	0	Any local user

To change the security of a file, enter SECURE followed by the name of the file, a comma, and a security string. For example, to assign the security string NONO to all the files in the subvolumes $FIDO.SPOT and $FIDO.ROVER, enter:

```
-SECURE ($FIDO.SPOT.*, $FIDO.ROVER.*), "NONO"
```

Instead of entering FUP codes, you could enter the program values in the security string. For example, you could enter 4242 instead of NONO. For more information about file security and security strings, see the description of the FUP SECURE command in the *File Utility Program (FUP) Reference Manual*.

Giving Files to Other Users

If you own a file, you can use the FUP GIVE command to give ownership of the file to another user (only one user can own a file). Enter GIVE followed by the name of a file, a file set, or a file-set list, a comma, and the user ID of the new owner.

For example, to give the files in the subvolume FORTLIB to user RESRCH.JACK, enter:

```
-GIVE FORTLIB.*, RESRCH.JACK
```

If you want to regain ownership of a file, the new owner or a user logged on as the super ID must use the FUP GIVE command to give the file to you.

Deleting Files

Use the FUP PURGE command to delete individual files or sets of files from your system. FUP prompts you for permission to purge files unless your command includes an exclamation point (!).

△ **Caution.** Be careful when you include the exclamation point (!) in a FUP PURGE command. The exclamation point means that the change you request will be made without further prompting. The results can be irreversible.

You must have purge access to a file in order to purge it. If you do not, you get purge error 48 (security violation).

Purging a Single File With or Without Prompting

To purge a single file and be prompted for permission:

1. Enter PURGE followed by the file name:

```
-PURGE OLDFILE
                CODE    EOF   LAST MODIF    OWNER  RWEP  TYPE  REC BLOCK
$DATA.RECDS
OLDFILE          101    848  17AUG92 12:22  8,44   NUNU  K     38   1024
PURGE?
```

2. To purge the file listed ($DATA.RECDS.OLDFILE), enter Y or y after the PURGE? prompt.

   If you press Return or type any character other than Y or y, FUP does not purge the file, and the FUP command prompt reappears.

   If you answer Y or y to the prompt, FUP purges the file and lists the name of the purged file:

```
-PURGE OLDFILE
                CODE    EOF   LAST MODIF    OWNER  RWEP  TYPE  REC BLOCK
$DATA.RECDS
OLDFILE          101    848  17AUG92 12:22  8,44   NUNU  K     38   1024
PURGE?y
$DATA.RECDS.OLDFILE PURGED.
```

To purge a file without being prompted for permission, add an exclamation point:

```
-PURGE OLDFILE !
$DATA.RECDS.OLDFILE PURGED.
1 FILE PURGED
```

Purging Several Files With or Without Prompting

You can specify more than one file or a file set in a PURGE command. When you specify several files, separate the elements with commas. For example, after you enter

this command, FUP prompts you for permission to purge each file named in the command:

```
-PURGE OLDFILE, NEWFILE, REDFILE
CODE   EOF   LAST MODIF   OWNER  RWEP TYPE REC BLOCK
$DATA.RECDS
OLDFILE 101  83465 17OCT85 14:32  8,44  CUCU
PURGE?y
$DATA.RECDS.OLDFILE PURGED.
CODE   EOF   LAST MODIF   OWNER  RWEP TYPE REC BLOCK NEWFILE 101 101388
15:18   8,44   CUCU
PURGE?n
CODE   EOF   LAST MODIF   OWNER  RWEP TYPE REC BLOCK
REDFILE 101   7754       12:03
PURGE?y
$DATA.RECDS.REDFILE PURGED.
2 FILES PURGED
```

If you add an exclamation point to your PURGE command, FUP purges the files with no further prompts. For example, after you enter the following command, FUP purges the files, informs you that the three files have been purged, and redisplays its command prompt:

```
-PURGE IDEAS.BAD, NOVELS.TRASH, NEWS.TABLOID !
$CORP.IDEAS.BAD PURGED.
$CORP.NOVELS.TRASH PURGED.
$CORP.NEWS.TABLOID PURGED.
3 FILES PURGED
```

Purging Subvolumes With Prompting

You can purge entire subvolumes of files by using the wild-card characters * or ? for the file name in a PURGE command. For example, you can purge all the files in the subvolume IDEAS by entering: -PURGE IDEAS.* After you enter a command to purge a subvolume, FUP displays a prompt asking if you want to purge the entire subvolume. The prompt lists four actions from which you choose:

```
-PURGE IDEAS.*
DO YOU WISH TO PURGE THE ENTIRE FILESET $CORP.IDEAS?
( Y[ES], N[ONE], S[ELECT], F[ILES] )?
```

You can choose to purge the subvolume with no more prompts (YES), to purge none of the files in the subvolume (NONE), to select the files to purge (SELECT), or to list the files in the subvolume (FILES). Type either the first letter of the word you choose or the entire word, and then press Return:

- If you choose YES, FUP purges the entire subvolume with no more prompting.
- If you choose NONE, the PURGE command ends without purging the subvolume. The FUP prompt reappears.

- If you choose SELECT, FUP displays information about each file in the subvolume and prompts you for permission to purge each file in turn:

```
-PURGE IDEAS.*
DO YOU WISH TO PURGE THE ENTIRE FILESET $CORP.IDEAS?
( Y[ES], N[ONE], S[ELECT], F[ILES] )?S
$CORP.IDEAS
CODE   EOF    LAST MODIF  OWNER  RWEP TYPE REC BLOCK
BAD     101 98732         11:15   8,44  AOGO
PURGE? Y
$CORP.IDEAS.BAD PURGED.
CODE   EOF    LAST MODIF  OWNER  RWEP TYPE REC BLOCK
GOOD    101  2048  18MAY87 9:10   8,44  CUCU
PURGE?N
CODE   EOF    LAST MODIF  OWNER  RWEP TYPE REC BLOCK
INDIFF  101 68129  03MAR87 8:46   8,44  CUCU
PURGE?Y
$CORP.IDEAS.INDIFF PURGED.
2 FILES PURGED
```

- If you choose FILES at the subvolume prompt, FUP lists the files in the subvolume, redisplays the PURGE command prompt, and waits for your next command:

```
-PURGE IDEAS.*
DO YOU WISH TO PURGE THE ENTIRE FILESET $CORP.IDEAS?
( Y[ES], N[ONE], S[ELECT], F[ILES] )?F

$CORP.IDEAS
    BAD    GOOD    INDIFF

DO YOU WISH TO PURGE THE ENTIRE FILESET $CORP.IDEAS?
( Y[ES], N[ONE], S[ELECT], F[ILES] )?
```

- If you press Return or type an answer that is not defined in the subvolume prompt, FUP behaves as if you answered SELECT.
- To exit the FUP PURGE command, press CTRL-Y.

Purging Subvolumes Without Prompting

If you place an exclamation point (!) at the end of your PURGE command, you can purge the entire subvolume of files with no further prompting:

```
-PURGE IDEAS.* !
$CORP.IDEAS.BAD PURGED.
$CORP.IDEAS.GOOD PURGED.
$CORP.IDEAS.INDIFF PURGED.
3 FILES PURGED
```

Using the NOPURGEUNTIL Option

If you want to prevent a file from being purged before a specific date, use the ALTER command with the NOPURGEUNTIL option to alter the file's expiration date. For example, this command prevents the file FILE1992 from being purged before January 1, 2000.

```
-ALTER FILE1992, NOPURGEUNTIL 1JAN2000,8:00:00
```

If you issue an INFO command with the DETAIL option, FUP displays the new expiration date:

```
-INFO FILE1992, DETAIL

$DISK2.ACCTS.FILE1992               15 Dec 1992, 11:48
ENSCRIBE
TYPE U
CODE 101
EXT ( 4 PAGES, 16 PAGES )
MAXEXTENTS 16
BUFFERSIZE 4096
OWNER 8,76
SECURITY (RWEP): NUNU
MODIF:   15 Dec 1992, 11:39
CREATION DATE:   4 Dec 1992, 10:39
LAST OPEN:   15 Dec 1992, 11:48
NOPURGEUNTIL:   1 Jan 2000,  8:00
EOF 4338 (0.9% USED)
EXTENTS ALLOCATED: 1
```

FILE1992 cannot be purged before 8:00 a.m. on January 1, 2000, except by a user with super ID 255,255 or unless you alter the expiration date again.

Using Your FUP Command History

FUP keeps a record of the commands you enter. The HISTORY command allows you to display your previous FUP commands. The FC command lets you change or correct any command in your FUP command history. These commands work similarly to the TACL commands of the same syntax described under Using Your Command History on page 2-15.

HISTORY Command

You can specify the number of previous FUP commands to display by entering a number after the command. If you do not enter a number, FUP displays the last ten commands. To display your last four FUP commands, enter:

```
- HISTORY 4

6: INFO MYFILE, DETAIL
7: CREATE NEWFILE
8: DUP OLDFILE, NEWFILE
9: HISTORY 4
```

FC Command

FC allows you to recall a command and then change it by deleting, inserting, or replacing characters. It can save you keystrokes. For example, to use the STATISTICS option of the INFO command and you have already used the DETAIL option, enter:

```
-FC 6
-INFO MYFILE, DETAIL
.             DDSTAT
-INFO MYFILE, STAT
.
```

! Command

The ! command allows you to reexecute a specific FUP command. For example, to reexecute the last INFO command, enter:

```
-! INFO
```

? Command

The ? command allows you to display (but not execute) a specific FUP command. For example, to display the last command that referenced the system \WEST, enter:

```
-? \WEST
```

Solving Common File Problems

Table 7-2 lists possible problems, causes, and solutions for common file problems. For information about database files, see Section 19, Monitoring Hardware Components.

Table 7-2. Common File Problems

Problem	Possible Causes	Solution
Information in data files becomes inconsistent.	Transaction failures, subsystem failures, disk failures, or operator error.	Escalate the problem; database recovery might be required. Refer to the *NonStop TM/MP Operations and Recovery Guide* for instructions on database recovery.
Transactions and queries directed at certain files are slow.	Too many transactions are directed at a single disk or file: an alternate key is needed.	Escalate the problem. Operations and database management can decide whether it is appropriate to move files, partition files, or create alternate key files or indexes.
Users or applications have problems accessing a certain file or files.	Failure to open an alternate key file (error 4).	Use the FUP commands ALTER or LOAD ALTFILE.
	File is not in the directory, or the record is not in the file (error 11).	Check syntax and reenter if necessary. Use the RESTORE utility to restore the file to disk.
	File is in use (error 12).	Wait and try later.
	File is full (error 45).	Make the file larger.
	A security violation occurred (error 48).	Check security setting and alter if needed.
	File is corrupted or bad (error 59).	Escalate the problem to your management.
A file or files are left open when the system is shut down.	Miscommunication, user forgetfulness, or faulty or incomplete system shutdown procedures.	Use the FUP LISTOPENS command to identify the user in question; notify your management.

8

Using FUP for Advanced File Management

This section contains information for users who are familiar with both the basic uses of the File Utility Program (FUP) and with Enscribe, the Compaq *NonStop*™ Kernel database record manager:

For more information about file structures in NonStop™ Kernel systems, see the *ENSCRIBE Programmer's Guide*.

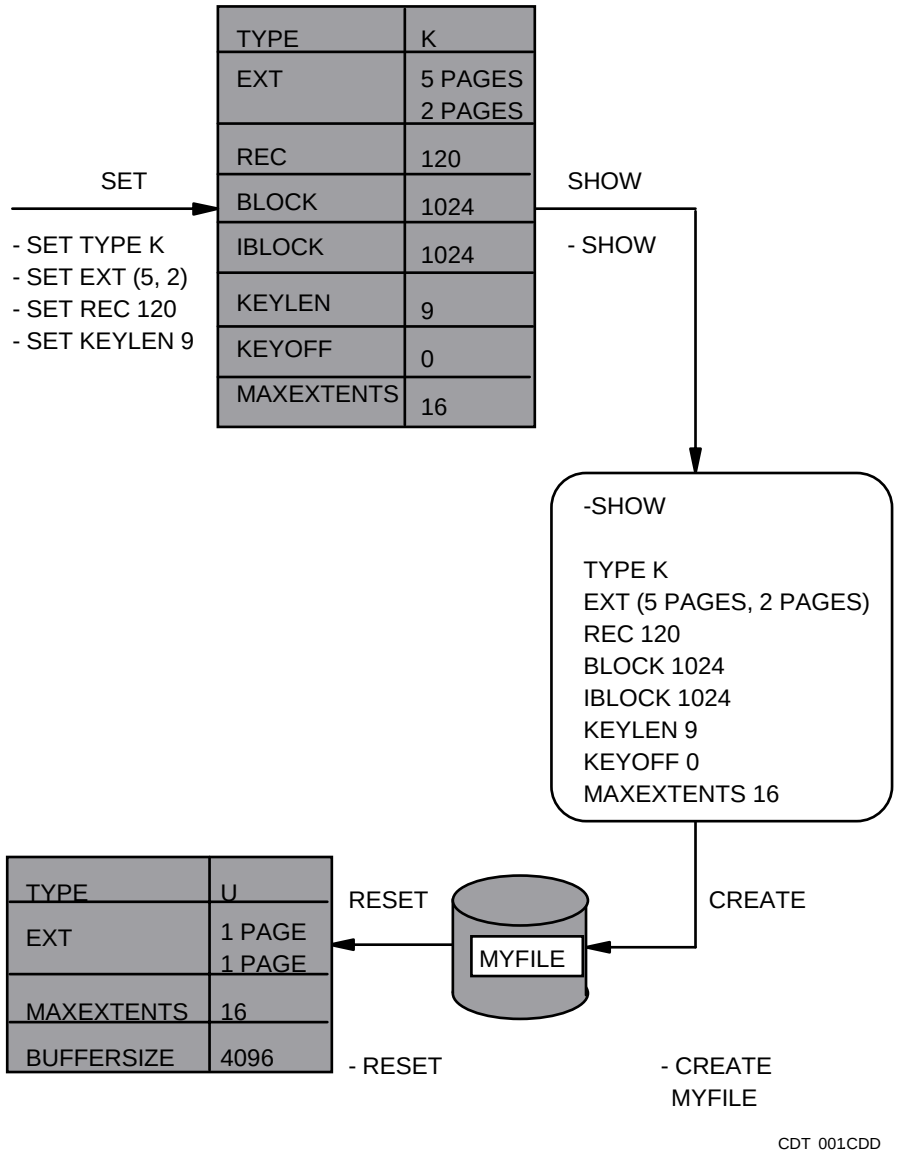
Creating Files

You can use FUP to create both structured and unstructured disk files. (You can create unstructured files with the TACL CREATE command. For information about the TACL CREATE command, see the *TACL Reference Manual*.)

To create a file with FUP:

1. Assign values to file-creation parameters with the FUP SET command. FUP maintains a table of current file-creation parameters. The values of these parameters can determine the attributes of any file you create with FUP. (You can also override the current settings by specifying different values in your CREATE command.)
2. Check the values of file-creation parameters with the FUP SHOW command to ensure that the values are correct, before creating a new file.
3. Create the file with the FUP CREATE command. When you enter the CREATE command, FUP consults its table of file-creation parameters and, if the current values result in a legal file, creates a file whose attributes are based on these values.
4. Restore one or more file-creation parameters to their original values with the FUP RESET command.

Figure 8-1. Steps for Creating a File With FUP



The file-creation parameters for the FUP SET command determine the default characteristics of the file you create. Use the FUP SHOW command to display the current values of the file-creation parameters. You can override the current value by setting a value for a parameter in your FUP CREATE command.

Table 8-1. Parameters of the FUP SET Command

Parameter	File Characteristic
TYPE	File type (unstructured, key-sequenced, relative, or entry-sequenced)
CODE	File code (to identify files)
EXT	Primary and secondary extent sizes
LIKE	Parameters to match an existing file
REFRESH	Automatic file-label refreshment
AUDIT	File auditing by TMF
REC	Record length
BLOCK	Data-block length
BLOCK	Index-block length
COMPRESS	Data and index compression
DCOMPRESS	Data compression
ICOMPRESS	Index compression
INSERTIONORDER	Insertion-ordered alternate key sequencing
KEYLEN	Primary-key length
KEYOFF	Primary-key offset
NO NULL	Null Value
QUEUEFILE	Queuefile attributes
UNIQUE	Unique-key specifications
PHYSVOL	Volume location specifications
ALTKEY	Alternate-key specifications
ALTFILE	File name of an alternate-key file
ALTCREATE	Automatic creation of alternate-key files
PART	Secondary-partition specifications
PARTONLY	Creation of individual partitions
ODDUNSTR	Reading and writing of odd-numbered bytes
MAXEXTENTS	Maximum number of file extents (DP2 files only)
BUFFERED	Mode of handling write requests (DP2 files only)
BUFFERSIZE	Size of internal buffer for unstructured files (DP2 files only)
AUDITCOMPRES	Mode of auditing by TMF (DP2 files only)
VERIFIEDWRITES	Mode for disc writes (DP2 files only)
SERIALWRITES	Selection of serial or parallel mirror writes (DP2 files only)

Creating Files Using DDL

You can also create structured files with the Tandem Data Definition Language (DDL). DDL is a data-management tool for describing large databases:

1. Describe the files in a DDL source schema.
2. Compile the schema with the DDL compiler to produce both a database dictionary and a file containing FUP file-creation commands.
3. Run FUP, specifying the command file created by DDL as an IN file.

For more information about DDL, see the *Data Definition Language (DDL) Reference Manual*.

Using the SET, SHOW, and CREATE Commands

When you start a FUP process, the default file-creation parameters are in effect. Use the FUP SHOW command to display the current values of the parameters. If you enter the SHOW command just after you start FUP, the initial default values are displayed:

```
10> FUP
-SHOW
    TYPE U
    EXT ( 1 PAGES, 1 PAGES )
    MAXEXTENTS 16
    BUFFERSIZE 4096
```

The TYPE parameter determines the file type of a file. The default TYPE value is U (for Unstructured). The EXT parameter determines the sizes of primary and secondary file extents; the default extent sizes are 1 page each (a page is a unit of 2048 bytes).

MAXEXTENTS and BUFFERSIZE appear for DP2 files only. MAXEXTENTS controls the maximum number of extents that can be allocated to a file. BUFFERSIZE sets the size in bytes of the buffer used for unstructured files. (For information about DP2 file attributes, see the *File Utility Program (FUP) Reference Manual*.)

If you enter the CREATE command at this point, you create an unstructured file whose primary and secondary extent sizes are one page each:

```
-CREATE FILE1
CREATED - $MANUF.FREDFILE.FILE1
```

To see the file attributes, enter the INFO command with the file name and DETAIL option:

```
-INFO FILE1, DETAIL
$MANUF.FREDFILE.FILE1           20 Jan 1992 11:04
    ENSCRIBE
    TYPE U
    EXT ( 2 PAGES, 2 PAGES )
    MAXEXTENTS 16
    BUFFERSIZE 4096
    OWNER 8,44
    SECURITY (RWEP): NUNU
    MODIF: 20 Jan 1992 11:03
    CREATION DATE: 20 Jan 1992 11:03
    LAST OPEN: NEVER OPENDED
    EOF 0 ( 0.0% USED)
    EXTENTS ALLOCATED: 0
```

To create a different type of file, such as a key-sequenced file, first specify the file type with the FUP SET command:

```
-SET TYPE K
```

For a key-sequenced file, you must also specify the length of the primary key:

```
-SET KEYLEN 9
```

Now if you enter the SHOW command, this information is displayed:

```
-SHOW
    TYPE K
    EXT ( 2 PAGES, 2 PAGES )
    REC 80
    BLOCK  4096
    IBLOCK 4096
    KEYLEN 9
    KEYOFF 0
    MAXEXTENTS 16
```

This SHOW display lists more attributes than those you assigned with your SET command. The additional values are defaults that apply to structured files (including key-sequenced files, files with alternate keys, and partitioned files).

Before you create a file, enter a FUP SHOW command and check that the values of the file-creation parameters are the ones you want. Correct any parameters you want to change, and then create the file.

Restoring Default File-Creation Parameters

After you create a file (and before you create another), you might want to restore certain file-creation parameters to their default values. To do this, enter the RESET command followed by the parameter or parameters to reset.

For example, you can reset more than one parameter with a single RESET command by separating the parameters with commas:

```
-SHOW <displays current values before you enter RESET
    TYPE K
    EXT ( 5 PAGES, 5 PAGES )
    REC 80
    BLOCK  4096
    IBLOCK 4096
    KEYLEN 9
    KEYOFF 0
    DCOMPRESS, ICOMPRESS
    MAXEXTENTS 16
-RESET EXT, COMPRESS
-SHOW  <displays new parameter values after RESET
    TYPE K
    EXT ( 1 PAGES, 1 PAGES )
    REC 80
    BLOCK  4096
    IBLOCK 4096
    KEYLEN 9
    KEYOFF 0
    MAXEXTENTS 16
```

To reset all file-creation parameters at once, enter RESET with no parameters. For example, this command restores all the file-creation parameters to their default values:

```
-RESET
-SHOW
    TYPE U
    EXT ( 1 PAGES, 1 PAGES )
    MAXEXTENTS 16
    BUFFERSIZE 4096
```

File-Creation Examples

These examples show how to create all the types of files: unstructured, entry-sequenced, relative, and key-sequenced files. Examples are also given for creating files with alternate keys, partitioned files, and files that match the attributes of an existing file. Each example shows the series of commands needed.

Creating an Unstructured File

Unstructured files are arrays of bytes. They are normally used to store object programs or text created with a text editing program such as TEDIT. If you create a file that has the logon default file-creation attributes, the new file is an unstructured file.

To create an unstructured file named $USERS.JOHN.UNSTRUCT whose primary extent size is 10 pages (20,480 bytes), whose secondary extent size is 2 pages (4096 bytes), and whose file code is 999, enter:

```
-SET EXT (10,2) --You can specify extent sizes in
                --pages, bytes, records, and
                --megabytes.  If you do not specify a
                --unit, FUP assumes that the unit is
                --pages.
-SET CODE 999   -- Set the file code (used to identify
                -- the file).
-SHOW           -- Show the current parameter values.
    TYPE U
    CODE 999
    EXT ( 10 PAGES, 2 PAGES )
    MAXEXTENTS 16
    BUFFERSIZE 4096
-CREATE UNSTRUCT    -- Create the file.
CREATED - $USERS.JOHN.UNSTRUCT
```

Creating an Entry-Sequenced File

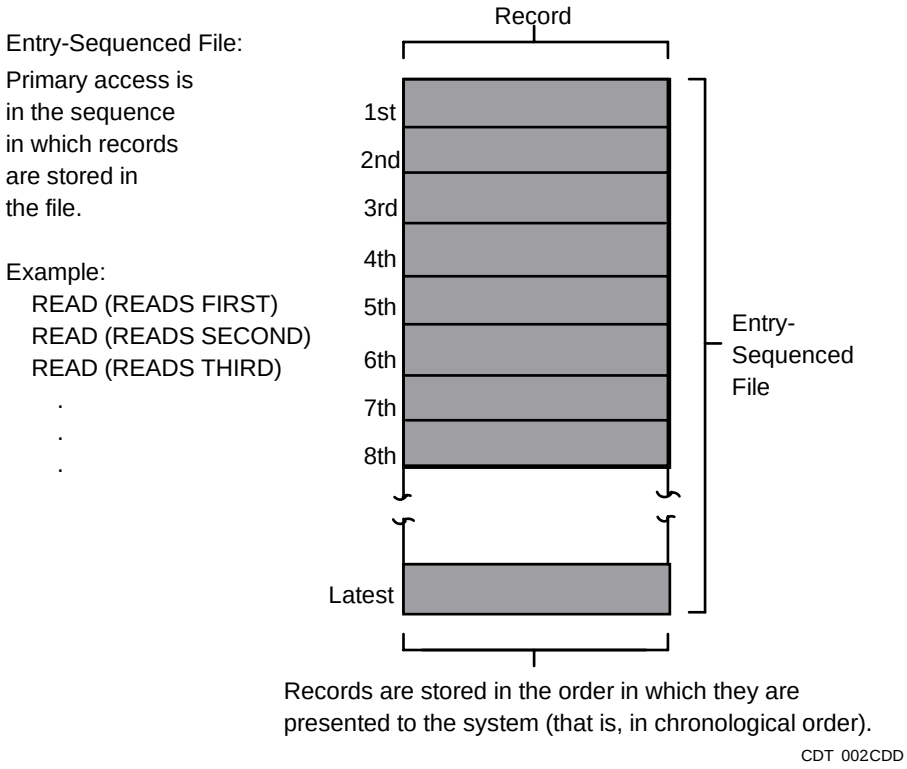
Entry-sequenced files have these characteristics:

- Records are searched sequentially from the beginning of the file.
- Records added to the file can vary in length, but once a new record is added, its length cannot change.
- Records in the file can be updated but not deleted.

To create an entry-sequenced file named $USERS.JOHN.ENSEQ whose primary extent and secondary extent sizes are four pages each, and whose data-block length is 2048 bytes, enter:

```
-SET TYPE E  --Set the file type to entry-sequenced.
-SET EXT 4   --You can specify extent sizes in pages,
             --bytes, records, and megabytes.  If you
             --do not give a unit, FUP assumes pages.
-SET BLOCK 2048   --Set the data-block length.
-SHOW             --Show the current parameter values.
    TYPE E
    EXT ( 4 PAGES, 4 PAGES )
    REC 80
    BLOCK  2048
    MAXEXTENTS 16
-CREATE ENSEQ       --Create the file.
CREATED - $USERS.JOHN.ENSEQ
```

Figure 8-2. Structure of an Entry-Sequenced File



Creating a Relative File

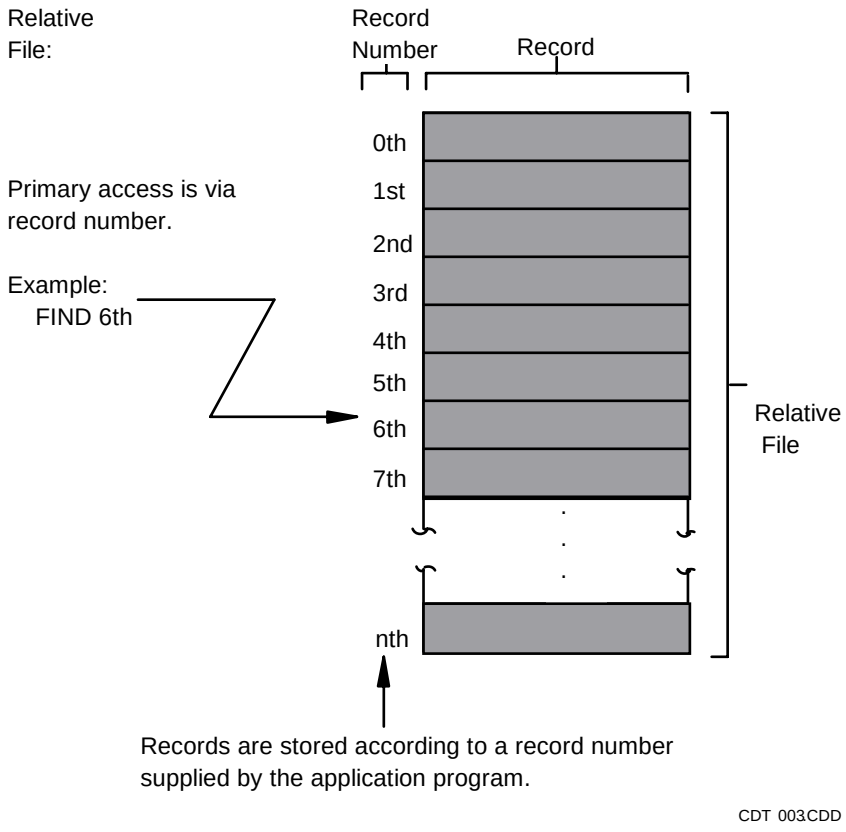
Relative files have these characteristics:

- All physical records are the same length.
- Records are stored by record number. Record numbers give the position of a record relative to the first record in the file.
- Records are retrieved randomly using record numbers.

To create a relative file named $USERS.JOHN.RELATIVE whose primary extent size is 5 pages, whose secondary extent size is 2 pages, and whose record length is 120 bytes, enter:

```
-SET TYPE R      --Set the file type to relative.
-SET EXT (5,2)   --You can specify extent sizes in
                 --pages, bytes, records, and megabytes
.                --If you do not give a unit, FUP
                 --assumes pages.
-SET REC 120     --Set the record length to 120 bytes.
-SHOW            --Show the current parameter values.
    TYPE R
    EXT ( 5 PAGES, 2 PAGES )
    REC 120
    BLOCK  4096
    MAXEXTENTS 16
-CREATE RELATIVE        --Create the file.
CREATED - $USERS.JOHN.RELATIVE
```

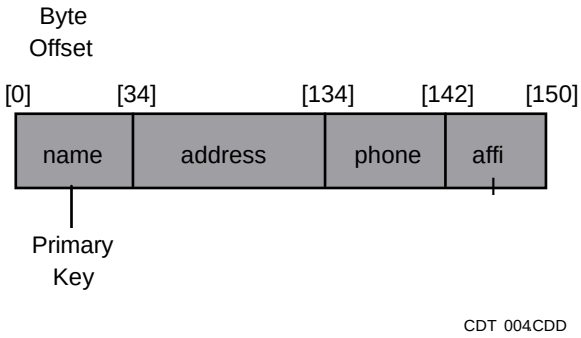
Figure 8-3. Structure of a Relative File



Creating a Key-Sequenced File

In a key-sequenced file, records are stored by the values of their primary keys. A primary key is a field within a record that uniquely identifies the record. Figure 8-4 shows a possible format for a record in a key-sequenced file.

Figure 8-4. Key-Sequenced File Format

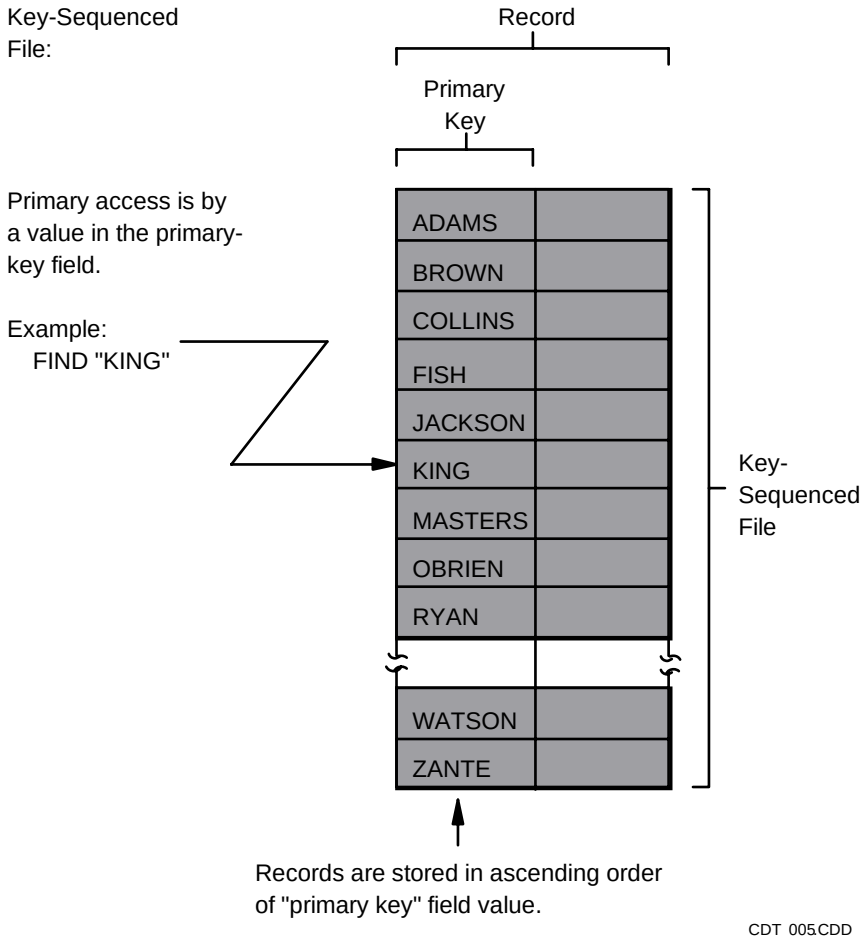


To create a file for records in the key-sequenced format, enter the following FUP commands in a disk file, and then start a FUP process that takes input from this file. This example shows the commands in FILE1.

```
20> TEDIT FILE1
VOLUME $VOL1.SVOL -- Set the default volume and
                  -- subvolume to the desired values.
SET TYPE K        -- Set file type as key-sequenced.
SET CODE 1000     -- You can optionally specify a file
                  -- code to identify the file.
SET EXT (16, 1)   -- Set primary and secondary
                  -- extent sizes for the application.
SET REC 150       -- Set record length and block size.
SET BLOCK 2048
SET COMPRESS      -- If desired, you can specify data
                  -- and index compression.
SET KEYLEN 34     -- You must specify a primary-key
                  -- length for key-sequenced files.
SET IBLOCK 2048   -- You can also specify the size of
                  -- index blocks.
SHOW              -- Display current parameter values.
CREATE KEYSEQ     -- Create the file.

21 > FUP /IN FILE1/
```

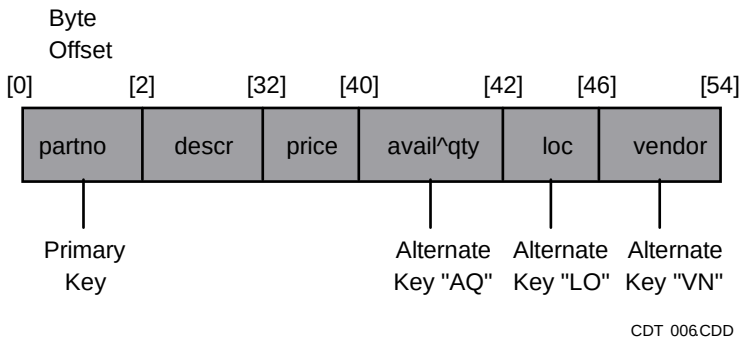
Figure 8-5. Structure of a Key-Sequenced File



Creating a Key-Sequenced File With Alternate Keys

Besides having a primary key, a key-sequenced file can have one or more alternate keys.

Figure 8-6. Possible Record Format: Key-Sequenced File With Alternate Keys



To create a file in key-sequenced format with alternate keys, first enter the following FUP commands in a file. Then run FUP, specifying the command file as the input file with the IN option. This example shows the commands in FILE2.

```
20> TEDIT FILE2
VOLUME $VOL1.SVOL  -- Set the default volume and
                   -- subvolume to the desired values.
SET TYPE K         -- Set file type to key-sequenced.
SET CODE 1001      -- Set optional file code to
                   -- identify the file.
SET EXT (32,8)     -- Set the appropriate primary and
                   -- secondary extent sizes for the
                   -- application.
SET REC 54         -- Set the record length.
SET BLOCK 4096     -- Set the block size.
SET IBLOCK 1024    -- Set the index block size.
SET KEYLEN 2       -- You must specify a primary-key
                   -- length for key-sequenced files.
--
-- Specify alternate keys and the name and number of
-- the alternate-key file.  If FILE is not specified
-- in the SET ALTKEY command, the alternate-key file
-- number is set by default to 0.
--
SET ALTKEY ("AQ", KEYOFF 40, KEYLEN 2)
SET ALTKEY ("LO", KEYOFF 42, KEYLEN 4)
SET ALTKEY ("VN", KEYOFF 46, KEYLEN 8)
SET ALTFILE (0, INVALT)
.
.
.
SHOW               -- Display current parameter values.
CREATE KEYSEQAL    -- Create the file.

21 > FUP /IN FILE2/
```

Creating a Key-Sequenced Partitioned File

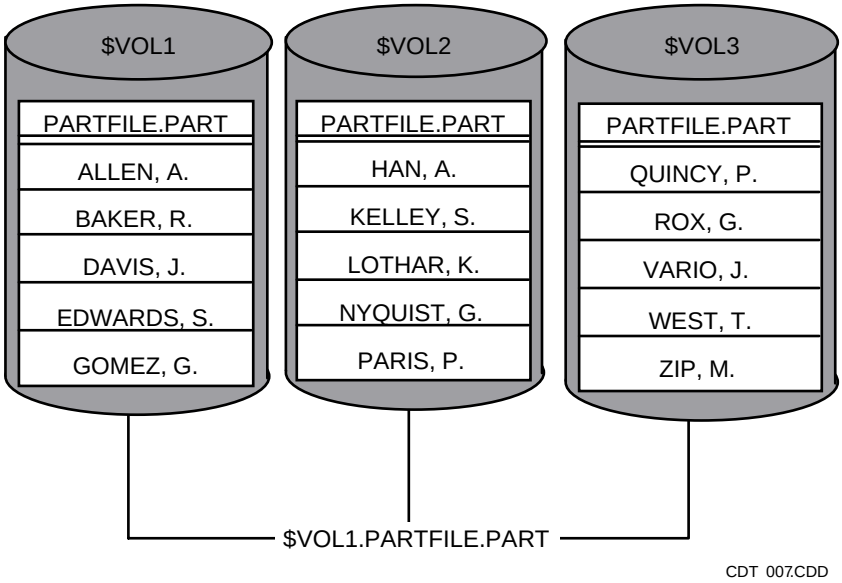
You do not need to keep all the data in a disk file on the same disk volume. By partitioning a file, you can store data in a file on as many as 16 different disk volumes. Partitioning allows you to create files that can be larger than those that reside on only one disk. Also, because the disk heads for each disk can be repositioned at the same time, access to data can be faster.

For example, suppose you want to create a partitioned file ($VOL1.PARTFILE.PART) whose primary partition resides in the disk volume $VOL1. You want the primary partition to contain the first record and all subsequent records up to, but not including, records whose primary key begins with HA.

You also want to create secondary partitions for the file. The first partition will reside on $VOL2, while the second will reside on $VOL3. The first secondary partition, on $VOL2, contains records whose primary key begins with HA and subsequent records up to, but not including, records whose primary keys begin with QU. All other records (that is, those records whose primary keys begin with QU and all subsequent records) will reside in the second secondary partition on $VOL3.

To create a file in key-sequenced partitioned format, enter the following FUP commands in a file. Then run FUP, specifying the command file as the input file with the IN option. This example shows the commands in FILE3.

Figure 8-7. Structure of a Partitioned File




```
20> TEDIT FILE3
VOLUME $VOL1.PARTFILE  -- Set the default volume
                       -- and subvolume .
SET TYPE K             -- Set the file type to
                       -- key-sequenced.
SET CODE 409           -- You can optionally specify a
                       -- a file code .
SET EXT (64,8)         -- Set appropriate primary and
                       -- secondary extent sizes for
                       -- the application.
SET REC 128            -- Set the record length.
SET KEYLEN 20          -- Specify the primary-key
                       -- length (required for
                       -- key-sequenced files).
--
-- Specify the partitions, their volumes, primary
-- and secondary extent sizes, and the partial-key
-- values
--
SET PART ( 1, $VOL2, 64, 8, "HA" )
SET PART ( 2, $VOL3, 64, 8, "QU" )
-- Show the current parameter
SHOW
CREATE PART            -- Create the file.

21 > FUP /IN FILE3/
```

Creating a File With the Attributes of an Existing File

If a file already exists that has all or most of the attributes you want a new file to have, use the LIKE option of the SET command to duplicate these attributes. To display the attributes of a file, enter the INFO command with the DETAIL option:

```
-INFO $CLEAN.BERKELEY.AIR, DETAIL
$CLEAN.BERKELEY.AIR                  10 Feb 1992 16:10
    ENSCRIBE
    TYPE K
    EXT ( 5 PAGES, 5 PAGES )
    REC 80
    BLOCK  2048
    IBLOCK 2048
    KEYLEN 9
    KEYOFF 0
    DCOMPRESS, ICOMPRESS
    MAXEXTENTS 16
    OWNER 8,44
    SECURITY (RWEP): NUNU
    MODIF:  25 Jan 1992 19:08
    CREATION DATE: 20 Jan 1992 12:09
    LAST OPEN: 22 Jan 1992 14:09
    EOF 0 (0.0 % USED)
    EXTENTS ALLOCATED: 0
    FREE BLOCKS 0
    INDEX LEVELS: 0
```

Make sure the parameters for the existing file have the values you want for the new file. Then enter the SET command with the LIKE option. FUP then sets its file-creation characteristics to match those of the file you specify. New files you create will have those attributes:

```
-SET LIKE $CLEAN.BERKELEY.AIR
-SHOW
    TYPE K
    EXT ( 5 PAGES, 5 PAGES )
    REC 80
    BLOCK  2048
    IBLOCK 2048
    KEYLEN 9
    KEYOFF 0
    DCOMPRESS, ICOMPRESS
    MAXEXTENTS 16
-CREATE $CLEAN.SANJOSE.AIR
CREATED - $CLEAN.SANJOSE.AIR
```

Maintaining Your Disk Files

You also use FUP to maintain your disk files after they are created:

All these operations involve changes in the file label of the affected file(s); in some cases, you must make the file-label changes yourself.

Each disk file has a file label — a record in the file directory that contains all the file attributes, such as the file name, file type, size of the file (EOF), extent size, names of partitions for partitioned files, and names of alternate-key files for key-sequenced files that have alternate keys.

Some file operations automatically change the file label. For example, when you add data to or delete data from a file, the EOF value in the file label changes accordingly. For other file operations, however, you must change file-label values yourself. For example, after you move an alternate-key file, you must change the name of the alternate-key file in the file label of the primary-key file.

To make changes to the file label, use the FUP ALTER command. Some examples in this section show file operations for which you must make file-label changes with the FUP ALTER command. The appropriate ALTER commands are included.

Loading Data Into Files

You can move data between files using these FUP commands:

- DUP — to duplicate an entire file (the original file and copy must be disk files).
- COPY — to move data one record at a time. COPY lets you copy records or part of a file to and from devices other than disks, including tape drives, printers, and terminals. COPY also lets you change file types by copying a file of one type into a target file of a different type.
- LOAD — to move data into a structured file. Data is transferred one record at a time from the source file and moved one block at a time into the destination file.

  The LOAD command has these advantages:

  - Loading files does not affect alternate-key values.
  - Because data is written a block at a time, LOAD is faster than COPY.

Using the LOAD Command

To load data into a file, enter LOAD followed by the name of the file that contains the data, a comma, and the name of the file to be loaded. This example shows how to load data stored on magnetic tape into a disk file. The name of the file to be loaded is:

```
$VOL1.SVOL.PART
```

PART is a key-sequenced file with three partitions. The secondary partitions are on volumes $VOL2 and $VOL3. The records to be loaded into this file are 128 bytes in length, on magnetic tape in unsorted order, with one record per block on the tape. The tape is mounted on the tape drive named $TAPE.

To load the file, enter:

```
-VOLUME $VOL1.PARTFILE.PART
-LOAD $TAPE, PART
```

When the LOAD command executes, the records from tape are first read, then sorted by primary key. After the records are sorted, FUP loads them into the partitions using the key specifications contained in the file label of PART. (Specify SORTED if the records are already sorted to avoid the time required to sort the file.)

You can include a number of options in a LOAD command. See the description of the LOAD command in the *File Utility Program (FUP) Reference Manual* for more information.

Purging Data From Files

Use the PURGEDATA command to purge data from a file without deleting the file. When you enter the PURGEDATA command, the end-of-file (EOF) pointer is set to 0, and other file-label values indicating the size of the file are reset to indicate that the file is empty. PURGEDATA does not, however, change the attributes of the file.

Enter PURGEDATA followed by a file name, a file set, or a file-set list. This example purges the data from a file set of two disk files:

```
-PURGEDATA ($VOL1.XDATA.FILE0, $VOL2.XDATA.FILE1)
```

Renaming and Moving Files With Alternate Keys

Renaming or moving a file that has alternate keys is complicated by the fact that the names of alternate-key files are attributes of the primary-key file. The file label of the primary-key file contains information about the alternate-key files, such as the file name and number specified as the ALTFILE attribute. When you rename or move either a primary-key file or an alternate-key file, you must record the change in the file label of the appropriate primary-key file.

To rename or move a file that has an alternate key:

- Use the FUP RENAME or DUPLICATE command to rename or move the file or files.

- Use the FUP ALTER command to change the file label of the primary-key file to reflect the change you made to the alternate-key file.

Renaming a File and Its Alternate-Key File

In this example, assume that you want to rename a structured file named $VOL1.SVOL.PRIFILE to the new name OLDSVOL.PRIFILE. The file has one alternate-key file named $VOL1.SVOL.AFILE, which you want to rename OLDSVOL.AFILE.

To change the names of the two files, enter:

```
-VOLUME $VOL1.OLDSVOL
-RENAME SVOL.PRIFILE,PRIFILE
-RENAME SVOL.AFILE,AFILE
```

To alter the file label of PRIFILE so that it includes the new name of the alternate-key file, enter:

```
-ALTER PRIFILE, ALTFILE ( 0, OLDSVOL.AFILE )
```

Moving a File With Alternate Keys to a New Volume

This example uses a structured file named $VOL1.SVOL.PRIFILE. It has one alternate-key file named $VOL1.SVOL.AFILE. You want to move the files to a new volume, $NEW, without changing the file names. To move them, enter:

```
-VOLUME $VOL1.SVOL
-DUP ( PRIFILE, AFILE ), $NEW.*.*
-VOLUME $NEW
-ALTER PRIFILE, ALTFILE ( 0, AFILE )
```

The ALTER command changes the name of the alternate-key file in the file label of PRIFILE from $VOL1.SVOL.AFILE to $NEW.SVOL.AFILE. After you complete the DUP and RENAME operations, you can purge the original file:

```
-PURGE $VOL1.SVOL.PRIFILE
```

Copying Files to a Backup Volume

Copying disk files to a backup disk volume gives you quick access to the file copies. However, you should use disk backups in addition to, not in place of, tape backups made with the BACKUP and RESTORE programs.

To perform a disk-to-disk backup operation, enter a FUP DUPLICATE command that copies all the files from one disk volume to another volume. For example, to duplicate all the files in disk volume $VOL1 to disk volume $BACKUP, enter:

```
-DUP $VOL1.*.*, $BACKUP.*.*, PARTONLY, SAVEALL
```

This command creates a backup copy of each file on $VOL1 on $BACKUP. Each backup file has the same file name and subvolume name as its original file. The

PARTONLY option means that only primary or secondary partitions that reside on $VOL1 are duplicated; partitions on other volumes are not copied. The SAVEALL option preserves the user ID, timestamp, and security setting for each file.

Adding Alternate Keys to Files

As your databases grow and change, you may need to add new keys to existing files. The steps for adding an alternate key to a file that already has alternate keys, and to a file that does not have alternate keys, are shown in the following examples.

Adding an Alternate Key in an Existing Alternate-Key File

For this example, suppose that you have a file named $VOL1.SVOL.PRIFILE. This file has an alternate-key file named $VOL1.SVOL.AFILE. You want to add the alternate-key records for the new key field to this file.

The key specifier for the new key is NM, the key offset in the record is 4, and the key length is 20. To add the new alternate key, enter:

```
-VOLUME $VOL1.SVOL
-ALTER PRIFILE, ALTKEY ( "NM", KEYOFF 4, KEYLEN 20 )
-LOADALTFILE 0, PRIFILE, ISLACK 10
```

The LOADALTFILE command loads the alternate-key records for key specifier NM and any other alternate keys into the alternate-key file. To allow for future growth of the file, you can reserve empty space in index blocks by specifying a percentage of slack space with the LOADALTFILE ISLACK option.

However, when you add a new alternate key, the length of the key can be no longer than the longest key already in the alternate-key file unless you:

- Use the SET LIKE command to duplicate the attributes of the old alternate-key file.
- Specify the new alternate key with the ALTER command.
- Specify a new record and key length for the alternate-key file that is 2 + length of the primary key + length of the longest alternate key.

For example, suppose that the file PRIFILE in the previous example has one alternate key that is 15 bytes long. The primary-key length is 40 bytes. The name of the alternate-key file is $VOL1.SVOL.ALTFILE. You can add a new alternate key, NM, with a length of 20 bytes by entering:

```
-VOLUME $VOL1.SVOL
-ALTER PRIFILE, ALTKEY ( "NM", KEYOFF 15, KEYLEN 20 )
-SET LIKE ALTFILE
-PURGE ALTFILE!
-SET REC 62
-SET KEYLEN 62
-CREATE ALTFILE
-LOADALTFILE 0, PRIFILE
```

Here, the new record and key lengths must be 62 bytes (2 for the key specifier, plus 20 for the longest alternate key, plus 40 for the primary key).

Adding an Alternate Key in a New Alternate-Key File

To add an alternate key to the file:

1. Create a new alternate-key file:

```
-VOLUME $VOL1.SVOL
-CREATE FILEB, TYPE K, REC 16, KEYLEN 16
-ALTER FILEA, ALTFILE (0, FILEB), ALTKEY ("XY", KEYLEN 10)
-LOADALTFILE 0, FILEA
```

   In this example, the new alternate-key file is named $VOL1.SVOL.FILEB. The key specifier for the new key is XY, the key offset in the record is 0, and the key length is 10. This example uses an entry-sequenced file named $VOL1.SVOL.FILEA, which does not have an alternate-key file

   The CREATE command creates the alternate-key file $VOL1.SVOL.FILEB. For non-unique alternate keys, the record length and key length are 16 bytes (2 for key specifier, plus 10 for the alternate-key field lengths, plus 4 for the primary key length). For unique alternate keys (specified by including the UNIQUE option in the SET ALTKEY command), the key length is 12 bytes (2 for key specifier, plus 10 for alternate-key field lengths), and the record length is 16.

   The ALTER command changes the file label for the primary-key file (FILEA) so that it specifies FILEB as the alternate-key file and contains the key specifier XY.

   Finally, the LOADALTFILE command loads the alternate-key records into the alternate-key file. Note that an index-block slack percentage of 0 is the default value.

Modifying Partitioned Files

Changing partitioned files involves making changes to specific partitions, then modifying the attribute of the file to reflect these changes.

Moving a Partition to a New Volume

This example uses a partitioned file named $VOL1.SVOL.PARTFILE. Secondary partitions of this file reside on volumes $VOL2 and $VOL3. To move the secondary partition on $VOL2 to the volume $NEW, enter:

```
-VOLUME $VOL1.SVOL
-DUP $VOL2.PARTFILE, $NEW.*, PARTONLY
-ALTER PARTFILE, PART ( 1, $NEW.SVOL.PARTFILE )
-PURGE $VOL2.PARTFILE
```

The DUP command with the PARTONLY option copies the secondary partition $VOL2.SVOL.PARTFILE to $NEW.SVOL.PARTFILE. The ALTER command changes the file label of the primary partition ($VOL1.SVOL.PARTFILE) to indicate that the first secondary partition resides in the file $NEW.SVOL.PARTFILE.

Loading a Partition of an Alternate-Key File

This example uses a key-sequenced, partitioned file named $VOL1.SVOL.PRIFILE. The file has alternate keys. The length of its primary-key field is 10. It has three alternate-key fields with key specifiers F1, F2, and F3. The length of each alternate-key field is 10 bytes.

All the alternate-key records are contained in one alternate-key file that is partitioned over three volumes. (To create such an alternate-key file, you must enter a SET NO ALTCREATE command to prevent automatic creation of an alternate-key file, and then create the partitioned alternate-key file separately.) Each volume contains the alternate-key records for one alternate-key field. This is possible because the key specifier for each alternate-key field is also the partial-key value for the secondary partitions.

The primary partition of the partitioned, alternate-key file is $VOL1.SVOL.AFILE. It contains the alternate-key records for the key specifier F1.

Partitions of the alternate-key file AFILE also reside in volumes $VOL2 and $VOL3. $VOL2.SVOL.AFILE contains the alternate-key records for the key specifier F2. $VOL3.SVOL.AFILE contains the alternate-key records for the key specifier F3.

To load the alternate-key records for the key specifier F2 into the file $VOL2.SVOL.AFILE, enter:

```
-VOLUME $VOL1.SVOL
-CREATE TEMP, EXT 30
-BUILDKEYRECORDS PRIFILE, TEMP, "F2", RECOUT 22, BLOCKOUT 2200
-LOAD TEMP, $VOL2.AFILE, PARTOF $VOL1, RECIN 22, BLOCKIN 2200
-PURGE ! TEMP
```

The CREATE command creates a temporary disk file to be used for output from the BUILDKEYRECORDS command. Next, BUILDKEYRECORDS generates alternate-key records that are to be loaded into the new file. The BUILDKEYRECORDS BLOCKOUT option specifies record blocking to improve the efficiency of disk write operations.

The LOAD command loads the secondary partition $VOL2.SVOL.AFILE. Because the SORTED option is not included, records are sorted before they are loaded. The RECIN option in the LOAD command specifies the same record blocking that was specified with the RECOUT option in the BUILDKEYRECORDS command.

Increasing the Extent Size of a Partition

This example uses a key-sequenced partitioned file ($VOL1.PARTFILE.PART) with secondary partitions in volumes $VOL2 and $VOL3. To increase the extent size of the partition in $VOL2, enter:

```
-VOLUME $VOL1.PARTFILE
-ALTER PART, PART ( 1, $VOL2, 120, 12 )
-RENAME $VOL2.PART, $VOL2.TEMP, PARTONLY
-SET LIKE $VOL2.TEMP
-SET EXT ( 120, 12 )
-CREATE $VOL2.PART
-DUP $VOL2.TEMP, $VOL2.PART, OLD, PARTONLY
-PURGE $VOL2.TEMP
```

The ALTER command with the PART option changes the file label of the primary partition so that it includes the new extent size of the secondary partition in $VOL2.

Next, the RENAME command with the PARTONLY option gives the secondary partition a temporary name; this command preserves the data in this partition.

The SET LIKE command recreates the file-creation parameters of the original secondary partition. Then the SET EXT command changes the extent size in the current FUP parameters. The CREATE command recreates the secondary partition with a larger extent size.

The DUP command with the OLD and PARTONLY options copies the data from the temporary file to the newly created partition with the DUP command. Finally, the PURGE command deletes the temporary file.

Although you can RENAME a file that is open for read-write or write-only access, you cannot copy such a file with the DUPLICATE command. Thus, you must ensure that the partition is not being written to if the preceding sequence of operations is to succeed. (File-access modes are discussed in the *ENSCRIBE Programmer's Guide*.)

Adding Partitions

You can add partitions to relative and entry-sequenced files that do not already have them, but not to key-sequenced files. This example uses a nonpartitioned relative file named $VOL1.SVOL.RELFILE. To add a partition to this file, enter:

```
-VOLUME $VOL1.SVOL
-SET LIKE RELFILE
-SET PARTONLY
-CREATE $VOL2.RELFILE
-SHOW EXT
    EXT ( 100 PAGES, 10 PAGES )
-ALTER RELFILE, PART ( 1, $VOL2 , 100, 10 )
```

The SET LIKE command sets the file-creation parameters to those of the original file, $VOL1.SVOL.RELFILE. The SET command with the PARTONLY option specifies that the file to be created is a secondary partition. The CREATE command creates a new partition on volume $VOL2.

The SHOW command with the EXT option displays the extent sizes of the original file. The ALTER command with the RELFILE option changes the file label of $VOL1.SVOL.RELFILE to show that it is the primary partition of a partitioned file with a secondary partition in the volume $VOL2.

This command adds a third partition on the volume $VOL3:

```
-CREATE $VOL3.RELFILE
-ALTER RELFILE, PART ( 2, $VOL3, 100, 10 )
```

Reorganizing Key-Sequenced Files

The FUP RELOAD command physically reorganizes a key-sequenced file or SQL object (table or index only) while allowing shared read/write access to the file or object. Reloading a file improves the access time and use of space for a key-sequenced file or SQL object that has undergone a large number of insertions, deletions, and updates. To reload a file called PAYFILE, enter:

```
-RELOAD PAYFILE, RATE 40
```

The RATE option controls the amount of processor time the reload operation uses. A value less than 100 prevents the reload operation from monopolizing the processor and its resources. For more information about FUP RELOAD and its other options, see the *File Utility Program (FUP) Reference Manual*.

Part III. Managing Disk and Tape Processes

This part of the guide contains information about tape activities, including labeled tape processing, and using Backup and Restore to copy files between disk and tape:

- Section 9, Performing Routine Disk Operations
- Section 10, Using Labeled Tapes
- Section 11, Backing Up and Restoring Disk Information

9

Performing Routine Disk Operations

This section describes how to perform routine operations related to the magnetic disks on your system:

Using the Subsystem Control Facility (SCF)

You perform many operations on disks and other peripheral devices through the Subsystem Control Facility (SCF). See the *SCF Reference Manual for G-Series Releases* for complete syntax rules, considerations, and examples for all SCF commands, along with SCF error messages.

Table 9-1. SCF Command Summary (page 1 of 4)

Command	Function
ABORT	(Does not support TAPE device in G-series releases) Stops the operation of an object without regard to the current state of its operation.
ACTIVATE	Returns a suspended object to the STARTED state.
ADD	Defines an object to a subsystem.
AGGREGATE	Displays the number of objects in the STARTED state and the STOPPED state, together with the total number of objects in a state other than STARTED or STOPPED.
ALIAS	Defines a command or text abbreviation.
ALLOCATE	Allocates (reserves) file space for one or more objects. Each object must be an explicitly (using the ADD command) or implicitly defined object.

Table 9-1. SCF Command Summary (page 2 of 4)

Command	Function
ALLOW	Specifies the number of errors or warnings permitted during the execution of a command file.
ALLOWOPENS	(Does not support TAPE device in G-series releases) Allows an object to once again accept opens; reverses the STOPOPENS command.
ALTER	Changes the value of one or more attributes of an object.
ASSIGN	Changes, adds, and displays ASSIGN messages passed to new processes initiated by the SCF RUN command.
ASSUME	Sets the default object type, object name, or both.
BOOT	Prepares the programmable controller for a software download. The controller should be in the STOPPED state for the BOOT command.
CHECK	Displays brief information about one or more objects. The display comprises abbreviated information from the INFO, NAME, STATS, and STATUS commands.
CLEAR	Clears the logical-file assignments made using the ASSIGN command, parameters set using the PARAM command, and aliases defined using the ALIAS command.
COMMENT	Allows the specification of descriptive text in command files.
CONFIRM	Enables or disables the display of positive responses from subsystems.
CONNECT	Establishes a connection between two objects.
CONTROL	(Does not support TAPE device in G-series releases) Allows the user to issue certain disk-specific commands (such as rebuilding the free-space table on magnetic disk).
COPY	Copies an object from one location to another.
CPUS	Displays the names of all known systems on the network and the states of their associated processors.
DEFAULT	Resets specific attributes to their default values.
DELAY	Suspends SCF processing for a specified time interval.
DELETE	Removes an object from the subsystem.
DETAIL	Enables or disables a labeled dump of the SPI command or response buffer, detailed error descriptions, and timestamp format and logging.
DIAGNOSE	Provides a means of diagnosing an object.
DISCONNECT	Breaks a connection between two objects.
DUMP	Copies information from one location to another to diagnose a problem.
ENV	Displays the settings of various SCF session parameters, sometimes called environmental parameters.
EXIT	Terminates an SCF session.

Table 9-1. SCF Command Summary (page 3 of 4)

Command	Function
FC	Allows correction of a previously entered SCF command line. (Fix command.)
HELP	Displays information about SCF commands.
HISTORY	Displays previously entered commands.
INFO	Displays system configuration information such as the current attribute values for a specified object.
INITIALIZE	(Does not support TAPE device in G-series releases) Prepares an object to be used in a Compaq *NonStop*™ Kernel system.
LISTDEV	Displays devices by name, type, or subsystem.
LISTOPENS	Displays the names of the processes that have the specified object open.
LISTPM	Displays process name, interface, and other information about the subsystem product modules.
LOAD	Loads a programmable controller with the program file specified in the ADD or ALTER command.
LOG	Directs to a file a copy of entered SCF commands and resulting displays.
MANAGERS	Displays information about SCP processes currently running.
MOVE	Moves (horizontally) an object and its subordinates to another location in a subsystem-defined object hierarchy.
NAMES	Displays the names described by an object-name template.
OBEY	Causes commands to be read from a file in noninteractive mode.
OPEN	Initiates communication with a specified SCP.
OUT	Redirects SCF displays.
PAGESIZE	Sets the terminal screen size and printer page size.
PARAM	Changes, adds, and displays PARAM values passed to a process initiated by the SCF RUN command.
PAUSE	Suspends SCF and allows the terminal to be used by another process. This command is usually used in conjunction with the RUN command.
PRIMARY	Causes the primary processor of a specified device to become the backup processor, and the backup processor to become the primary processor.
PROBE	Verifies that the link between two or more objects is operational.
RELEASE	Releases, or returns, an object's file space for reuse by the subsystem.
RENAME	(Does not support TAPE device in G-series releases) Changes the name of an object. You can use the RENAME command to resolve duplicate disk name conflicts; it lets you rename and bring up a volume that has the same name as another volume currently active on the system.
REPEAT	Causes the remainder of the command line to be repeated for a specific number of iterations or until the Break key is pressed.

Table 9-1. SCF Command Summary (page 4 of 4)

Command	Function
RESET	Puts an object in a state from which it can be started. The RESET command moves an object to a STOPPED state, substate DOWN from any state except the STARTED state (in which state RESET is ignored).
RUN	Runs another program from within SCF.
SAVE	Archives a copy of the CONFIG configuration database file.
SETMANAGER	Allows the explicit setting of a subsystem manager process name.
SETPROMPT	Changes the information contained in the command-line prompt.
START	Initiates the operation of an object (makes a stopped device accessible to user processes). If successful, leaves the object in a STARTED state.
STATS	(Does not support TAPE device in G-series releases) Displays the accumulated statistics for an object and optionally resets them.
STATUS	Displays current status information about an object.
STOP	Terminates access to a storage device in an orderly manner. The device is not stopped until current activity ends. When the STOP command is done, configured devices are left in a STOPPED state, substate DOWN. Devices remain configured in the system configuration database.
STOPOPENS	(Does not support TAPE device in G-series releases) Prevents any additional opens to an object.
SUSPEND	Restricts the use of the connections both to and from an object. Application requests, except close requests, are rejected. The SUSPEND command places the object into the SUSPENDED state.
SWITCH	(Does not support TAPE device in G-series releases) Causes the primary path to become the backup path, and the backup path to become the primary path. For disks, the SWITCH command designates the preferred ServerNet addressable controller (SAC) path for any magnetic disk accessible through dual paths.
SYSTEM	Sets the default system name for all file-name and object-name expansion.
TELL	Sends a text-string message to a subsystem.
TIMEOUT	Allows the user to vary the amount of time SCF waits for a response from SCP before canceling the request.
TRACE	Captures trace information (activity) of one or more selected objects.
VERIFY	Tests an object for its ability to meet an object-specific subsystem quality control objective.
VERSION	(Does not support DISK or TAPE devices in G-series releases) Displays the version level of SCF, SCP, or the current subsystem.
VOLUME	Sets default volume and subvolume names for all file-name expansion.
!	Reexecutes a command line.
?	Displays a specific command line.

Checking Disk Status

To list and check the current status of the disks on your system, enter:

```
> SCF STATUS DISK $*
```

A listing similar to this is sent to your home terminal:

```
29-> STATUS DISK $*
STORAGE - Status DISK \SHARK.$DATA02
LDev   Primary    Backup     Mirror     MirrorBackup   Primary   Backup
                                                       PID       PID
  62   STOPPED    STOPPED    STOPPED    STOPPED         0,268     1,265

STORAGE - Status DISK \SHARK.$DATA09
LDev   Primary    Backup     Mirror     MirrorBackup   Primary   Backup
                                                       PID       PID
  55   STOPPED    STOPPED    STOPPED    STOPPED         3,265     2,260

STORAGE - Status DISK \SHARK.$DATA10
LDev   Primary    Backup     Mirror     MirrorBackup   Primary   Backup
                                                       PID       PID
  54   STOPPED    STOPPED    STOPPED    STOPPED         3,264     2,261

STORAGE - Status DISK \SHARK.$DATA01
LDev   Primary    Backup     Mirror     MirrorBackup   Primary   Backup
                                                       PID       PID
  63  *STARTED    STARTED   *STARTED    STARTED         0,267     1,266

STORAGE - Status DISK \SHARK.$DATA04
LDev   Primary    Backup     Mirror     MirrorBackup   Primary   Backup
                                                       PID       PID
  60  *STARTED    STARTED   *STARTED    STARTED         0,270     1,263
    …
```

This listing shows that some of the disks are up and running normally and that others are stopped. See Table 19-2, SCF Object States, on page 19-6 for information on the different device states that can appear in the STATE column of the STATUS display.

Note. To obtain device listings for device types, subtypes, and other device characteristics, use the SCF STATUS options. See the *SCF Reference Manual for the Storage Subsystem* .

Examples

- To list of the device characteristics and status of the disk $SPOOL, enter:

  ```
  > SCF STATUS DISK $SPOOL
  ```

  A listing similar to this is displayed on your home terminal:

```
STORAGE - Status DISK \SHARK.$SPOOL
LDev   Primary    Backup     Mirror     MirrorBackup   Primary   Backup
                                                       PID       PID
   7  *STARTED    STARTED   *STARTED    STARTED         0,267     1,266
```

  This display shows that all paths to this mirrored volume are currently up.

- To list the devices on your default system, enter:

  ```
  > SCF STATUS $SYSTEM
  ```

A listing similar to this is sent:

```
STORAGE - Status DISK \SHARK.$SYSTEM
LDev   Primary    Backup     Mirror     MirrorBackup   Primary   Backup
                                                       PID       PID
   6  *STARTED    STARTED    STOPPED    STOPPED         0,256     1,256
```

This display shows that the primary disk is up but the mirror disk is down.

Bringing Up a Disk or Path

At various times, typically after maintenance, you will need to bring up a disk or path from a down, hard down, or exclusive state. See Section 19, Monitoring Hardware Components, for an explanation of device states.

Note. Before bringing up a disk that contains an SQL file, see your manager or database administrator.

Bringing Up a Disk From STOPPED

1. Make sure you are logged on as a super-group user (255,*n*).
2. Enter:

```
> SCF
-> START DISK $disk
```

Example

To bring the disk $DATA back up from a regular STOPPED state, log on as a super-group user (255,*n*) and enter:

```
> SCF START DISK $DATA
```

Bringing Up a Path From STOPPED

Enter:

```
> SCF
-> START DISK $disk { -P | -B | -M | -MB }
```

You must specify one of the following parameters for mirrored disks: -P (for primary path); -B (for backup path); -M (for mirror path); or -MB (for mirror backup path).

Bringing Up a Disk or Path From a STOPPED State, Substate HARDDOWN

1. Make sure you are logged on as a super-group user (255,*n*).
2. Reset the disk:

```
-> RESET DISK $disk
```

3. Bring the disk or path back up:

   ```
   -> START DISK $disk
   ```

Example

To reset the disk $DATA from a STOPPED state, substate HARDDOWN:

1. Make sure you are logged on as a super-group user (255,*n*).
2. Enter:

   ```
   -> RESET DISK $DATA
   ```

Taking Down a Disk or Path

You will need to take down a disk or path when:

- It is known to be malfunctioning
- The disk has been physically removed from the system and the system has not been reconfigured
- A diagnostic program is to be run on the disk.

△ **Caution.** Before bringing down $SYSTEM, see your operations manager or database administrator. Also, before you bring down a disk that contains an SQL file, check with your management.

Taking Down a Disk

1. Make sure you are logged on as a super-group user (255,*n*).
2. Stop the disk:

   ```
   > SCF
   -> STOP DISK $disk
   ```

3. Check the disk status to confirm it is down:

   ```
   > SCF STATUS DISK $disk
   ```

Example

To make the disk $AMBER inaccessible to user processes:

1. Make sure you are logged on as a super-group user (255,*n*).
2. Take the disk down:

   ```
   > SCF STOP DISK $AMBER, FORCED
   ```

   FORCED means that SCF continues even if there are open files on the disk.

3. Check the disk status to confirm it is down:

   ```
   > SCF STATUS DISK $AMBER
   ```

   A listing similar to this is displayed on your home terminal:

```
STORAGE - Status DISK \SHARK.$AMBER
LDev   Primary    Backup     Mirror     MirrorBackup   Primary   Backup
                                                       PID       PID
  61   STOPPED    STOPPED    STOPPED    STOPPED         0,269     1,264
```

Taking Down a Path To a Mirrored Disk

1. Make sure you are logged on as a super-group user (255,*n*).
2. Stop the path:

   ```
   > SCF
   ```

   ```
   -> STOP DISK $disk { -P | -B | -M | -MB }
   ```

   You must specify one of the following parameters for mirrored disks: -P (for primary path); -B (for backup path); -M (for mirror path); or -MB (for mirror backup path).

   If you do not specify a path, the following message is displayed:

```
STORAGE W01007 The STOP command will make all paths to $disk inaccessible.
```

3. Check the path status to confirm it is down:

   ```
   > SCF STATUS DISK $disk
   ```

Example

To make the primary path to the mirror disk $DATA inaccessible:

1. Make sure you are logged on as a super-group user (255,*n*).
2. Stop the path:

   ```
   > SCF STOP DISK $DATA-M
   ```

3. Check the path status to confirm it is down:

   ```
   > SCF STATUS DISK $DATA
   ```

   A listing similar to this is displayed on your home terminal:

```
STORAGE - Status DISK \SHARK.$DATA
LDev   Primary    Backup     Mirror     MirrorBackup   Primary   Backup
                                                       PID       PID
  50   STARTED    STARTED    STOPPED    STOPPED         3,260     2,265
```

Altering the Current Path to a Dual-Ported Disk

To test alternate paths to a dual-ported disk, you alter the current path:

1. Make sure you are logged on as a super-group user (255,*n*).
2. Enter the following form of the SCF SWITCH command:

   ```
   -> SWITCH $disk { -P | -B | -M | -MB }
   ```

   You must designate whether you want the primary (-P), backup (-B), mirror (-M), or mirror backup (-MB) controller path to become the current path to the disk. The path you specify then becomes the path of first choice when a user process needs to access the disk.

Example

In the partial STATUS DISK shown below, the STARTED states indicate that the primary and mirror paths of $AMBER are active:

```
1-> STATUS DISK $AMBER
STORAGE - Status DISK \SHARK.$AMBER
LDev   Primary    Backup     Mirror     MirrorBackup   Primary   Backup
                                                       PID       PID
   6   STARTED    STARTED    STARTED    STARTED         0,268     1,265
```

Removing Half of a Mirrored Disk

You might need to physically remove half of a mirrored disk to use for backups or other purposes.

1. Make sure you are logged on as a super-group user (255,*n*).

   Note. If the target disk contains any SQL file, see your operations manager or database administrator before issuing an SCF STOP DISK command on the target disk.

2. Bring the application to a quiet state to ensure the integrity of the files. This does not require that application programs close their files, but that the current transactions are completed normally.
3. If the target disk is protected by the Transaction Management Facility (TMF) subsystem, disable it from the TMF subsystem before performing this operation.
4. Issue an SCF STOP DISK command.

   SCF STOP DISK is valid only for physically demountable mirrored disks, and it is rejected if either disk of the mirrored disk is already down.

5. Determine whether any unspared bad sectors exist on the disk:

   ```
   -> INFO DISK $disk, BAD
   ```

6. Spare any bad sectors:

   ```
   -> CONTROL DISK $disk { -P | -M }, SPARE
          , cylinder , head , sector
   ```

   where *cylinder*, *head*, and *sector* are the numbers of the cylinder, head, and sector to be assigned an alternate sector. Do not spare cylinder 0, or head 0, sector 0 on any disk; doing so makes the disk unusable. Do not spare sectors 0, 1, 2, or 3 of cylinder 0, head 0 on the disk $SYSTEM; doing so prevents you from loading the system from that disk.

   The SCF INFO DISK, BAD and CONTROL DISK, SPARE commands are described in detail in the *SCF Reference Manual for the Storage Subsystem*.

7. Stop the paths to the disk half:

   ```
   -> STOP DISK $disk -M, FORCED
   ```

   ```
   -> STOP DISK $disk -MB, FORCED
   ```

   FORCED means that SCF continues even if there are open files on the disk.

   A warning message and verification request is displayed.

8. Enter Y to continue. STOP DISK brings down the device on which the disk is mounted.

9. Issue the SCF STATUS DISK command and check to verify that the STOP command put the specified disk in the STOPPED state.

10. Power off the disk in TSM. For information on using TSM, see the *TSM Online User's Guide.*

11. When the disk stops spinning, physically remove the disk pack.

Example

To remove the primary disk of $DATA00 with SCF before you physically remove it:

1. Make sure you are logged on as a super-group user (255,*n*).

2. Enter:

   ```
   -> INFO DISK $DATA00, BAD
   ```

   ```
   -> CONTROL DISK $DATA00-P, SPARE 173, 2, 6
   ```

   ```
   -> STOP DISK! $DATA00-P
   ```

   A warning message and verification request is displayed.

3. Respond yes by entering:

   ```
   -> Y
   ```

4. Check the status of $DATA00:

   ```
   -> STATUS DISK $DATA00
   ```

```
STORAGE - Status DISK\ALM171.$DATA00
LDev   Primary    Backup    Mirror    MirrorBackup   Primary   Backup
                                                    PID       PID
 6     STOPPED    STOPPED   STARTED   STARTED         0,10      1,10
```

   The STOPPED status in this listing confirms that the paths to the primary disk $DATA00-P and $DATA00-B are now down.

5. Stop the drive.
6. When the disk stops spinning, physically remove the disk pack.

Bringing Up the Down Half of a Mirrored Disk

SCF determines when a revive is needed and prompts you. If a disk is part of a mirrored volume and half of the disk pair is already up, bring up the downed half of the mirrored disk into the STARTED state:

1. Make sure you are logged on as a super-group user (255,*n*).
2. Start the SCF program:

   ```
   > SCF
   ```

3. Check for bad sectors and checksum errors on $*disk*:

   ```
   -> INFO DISK $disk, BAD
   ```

4. If there are errors, correct them.
5. Perform the START DISK operation:

   ```
   -> START DISK $disk
   ```

   A warning message and verification request is displayed:

6. Enter Y to continue.

   The START DISK operation copies data from the up half to the down half while concurrent processing on the up half continues.

   To check the status of the operation:

   ```
   -> STATUS DISK $disk , DETAIL
   ```

Example

This example shows how to revive a mirrored disk, half of which has been physically removed for use as a disk backup and replaced with a new disk pack. The mirrored disk $DATA01 is physically demountable and an SCF STATUS DISK command shows both $DATA01-M and $DATA01-MB to be down.

1. Check for bad sectors and checksum errors on $DATA01:

   ```
   -> INFO DISK $DATA01, BAD
   ```

```
STORAGE - Bad Sector Information DISK\ALM171.$DATA01
 No bad sectors found.
```

2. Log on as a super-group user (255,*n*), enter SCF, and logically remove $DATA01-M from the system:

   ```
   -> STOP DISK $DATA01-M
   ```

   ```
   -> STOP DISK $DATA01-MB
   ```

3. Physically remove the $DATA01-M disk. Put it on a storage shelf.
4. Mount a previously formatted disk.
5. Start the newly installed disk pack from the primary of $DATA01 so that the new pack becomes the mirror disk of $DATA01:

   ```
   -> START DISK $DATA01
   ```

   SCF prompts you to start a revive operation. Enter Y if you want to continue.

   To check the progress of the start operation, enter:

   ```
   -> STATUS DISK $DATA01, DETAIL
   ```

When the START DISK operation finishes, the mirror device is in the STARTED state.

Finding and Sparing Bad Tracks and Sectors

When a disk pack is initially formatted, a fixed number of sectors are reserved at the end of each cylinder as spares. This number varies depending on the type of disk. These spare sectors are used only if defective sectors are found on the cylinder.

Sparing sectors is the process of assigning alternate sectors from the available spare sectors to be used in place of defective sectors, removing associated unspared bad sector entries from the sector information table (if necessary), and creating entries in the added defect map. You will only need to spare defective sectors that were not automatically spared by the formatting operation.

1. List all of the defective sectors on the disk to determine which sectors, if any, need to be spared:

   ```
   > SCF
   ```

   ```
   -> INFO DISK $disk, BAD
   ```

   SCF displays a listing that gives you the logical cylinder, head, and sector address of any bad sectors. If there are no bad sectors, there is no need to continue with this procedure.

2. Make sure you are logged on as a super-group user (255,*n*).
3. Stop all other processing on the disk.

4. Make sure there is not already an SCF CONTROL DISK, SPARE operation in progress. (You must do this yourself; SCF does not warn you if you start a second CONTROL DISK, SPARE operation.)
5. Spare the defective sectors:

   ```
   > SCF
   -> CONTROL DISK $disk, SPARE sector-address
   ```

   sector-address must be specified and includes the numbers of the cylinder, head, and sector to be assigned an alternate sector.

△ **Caution.** Do not spare cylinder 0, or head 0, sector 0 on any disk; doing so makes the disk unusable. Do not spare sectors 0, 1, 2, or 3 of cylinder 0, head 0 on the disk $SYSTEM; doing so prevents you from loading the system from that disk.

For more information about using these SCF commands, and their options, see the *SCF Reference Manual for the Storage Subsystem*.

Example

1. List the defective sectors on the disk, $DATA01:

   ```
   > SCF
   -> INFO DISK $DATA01, BAD
   ```

   A listing similar to this is sent to your home terminal:

```
STORAGE - Bad Sector Information DISK \ALM171.$DATA01

Bad Sectors Information $DATA01 Primary:

No bad sectors information is available.

Bad Sectors Information $DATA01 Mirror:

Logical Sector Address                    Date Detected
%H0000795C                                SEPTEMBER 22, 1996 15:25:12

File Name               File Address         Logical Sector Address
DLSYS42X.OSIMAGE         573440-573951        %H0000795C
```

   This listing shows sectors on this disk that the disk process has flagged as bad.

2. Make sure you are logged on as a super-group user (255,*n*).
3. Stop all other processing on the disk.
4. Make sure there is not already an SCF CONTROL DISK, SPARE operation in progress.
5. Unconditionally spare the sector that has logical sector address %H0000795C:

   ```
   > SCF
   -> CONTROL DISK $DATA01, SPARE %H795C
   ```

Managing Disk Space Usage

You should regularly monitor the amount of available space on a system to ensure that problems do not arise for users because of a lack of available disk space.

Disks, unlike other system resources, are subject to performance degradation simply through normal usage. The addition and deletion of records and files can cause file and disk fragmentation over time. Adding large amounts of data to a file can cause inefficient internal file structures.

Disks can also become full, which prevents system users from creating additional files, subvolumes, or volumes.

To make more disk space available to users:

1. Analyze your current available disk space.
2. Create more disk space as needed based on your analysis, by either using DCOM to compress existing files and subvolumes, or by purging old and unneeded files and subvolumes.

You can analyze disk space using either the Subsystem Control Facility (SCF) or the Disk Space Analysis Program (DSAP).

Analyzing Disk Space Usage With the Subsystem Control Facility (SCF)

The SCF STATUS DISK, DETAIL command lists the number of free pages, the sizes of contiguous blocks of free pages, and the number of files currently on a specified disk. It also lists the number of pages in the primary extent of the disk directory, and the number of extents that have been allocated for the disk directory.

To check the free space on a disk with the SCF STATUS DISK, DETAIL command, enter:

```
> SCF STATUS DISK $disk,  DETAIL
```

The format of the resulting display and details about this command are described in the *SCF Reference Manual for the Storage Subsystem.*

Example

To display disk free-space information for the disk $AMBER, enter:

```
> SCF STATUS DISK $AMBER, DETAIL
```

A report such as this is returned to your home terminal:

```
STORAGE - Detailed Status DISK \SHARK.$AMBER

Disk Path Information:
 LDev  Path           PathStatus  State      SubState         Primary   Backup
                                                              PID       PID
   63  PRIMARY        ACTIVE      STARTED                       0,267     1,266
   63  BACKUP         INACTIVE    STARTED                       0,267     1,266
   63  MIRROR         ACTIVE      STARTED                       0,267     1,266
   63  MIRROR-BACKUP  INACTIVE    STARTED                       0,267     1,266

General Disk Information:
  Device Type........... 3           Device Subtype........... 40
  Primary Drive Type.... 4565-1      Mirror Drive Type........ 4565-1
  Physical Record Size.. 4096        Priority................. 220
  Library File..........
  Program File.......... \SHARK.$SYSTEM.SYS00.TSYSDP2
  Protection............ MIRRORED

Usage Information:
  Capacity (MB)......... 2000.09     Free Space (MB).......... 290.76 (14.53%)
  Free Extents.......... 16          Largest Free Extent (MB). 172.42

General Disk Information:
  Device Type........... 3           Device Subtype........... 40
  Primary Drive Type.... 4565-1      Mirror Drive Type........ 4565-1
  Physical Record Size.. 4096        Priority................. 220
  Library File..........
  Program File.......... \SHARK.$SYSTEM.SYS00.TSYSDP2
  Protection............ MIRRORED

Usage Information:
  Capacity (MB)......... 2000.09     Free Space (MB).......... 290.76 (14.53%)
  Free Extents.......... 16          Largest Free Extent (MB). 172.42

Hardware Information:
  Path            Location              Power           Physical Status
                  (group,module,slot)
  PRIMARY         (1,1,3)               DUAL            PRESENT
  MIRROR          (1,1,4)               DUAL            PRESENT
```

Analyzing Disk Space Usage With the Disk Space Analysis Program (DSAP)

The Disk Space Analysis Program (DSAP) analyzes and displays how disk space is being used on a specified disk. It copies the disk directory and the disk free-space table to its own working storage and then, depending on which options you specify, manipulates this data to produce reports relevant to the use of disk space on that disk.

DSAP only measures the use of disk space; it makes no modifications to your system. DSAP is a privileged program and might not be available to all users on your system. If you have questions about your access to DSAP, check with your system manager.

DSAP is a noninteractive program that runs online while your disk is up. You run DSAP by entering a DSAP command with parameters at a TACL prompt. DSAP then takes control of your terminal and generates the requested report. If you enter DSAP at a TACL prompt without any parameters, DSAP displays help information, including run options, report options, and examples.

To use DSAP to quickly check the free space on a disk, enter:

```
> DSAP $disk, SHORT
```

You can exit DSAP by pressing the Break key, but DSAP continues to run in the background until it finishes. To continue running DSAP in the foreground after pressing Break, enter PAUSE at the TACL prompt. To stop DSAP entirely, you can enter STOP at the TACL prompt if DSAP was the last process you started.

DSAP can produce nine different report types, each of which can have several selection options. Some of these reports are useful only to system operators, system managers, or group managers. There are three reports that you might find useful:

Report Type	Report Description	Usage Example
Summary Report	A summary of your space for a specific disk volume; includes your total number of allocated page, extents, unused pages, number of files, and SQL views.	To generate, enter: `> DSAP $disk , BYUSER`

Subvolume Summary Report	A Summary Report and analysis of disk use for each subvolume on the disk. If you specify a user ID, the report contains information only for that user's subvolumes.	To generate, enter:
		`> DSAP $disk , BYSUBVOL`
		To generate for a specific user, enter:
		`> DSAP $disk , BYSUBVOL, USER group-id,user-id`

User Detail Report	A Summary Report and detailed information about each of your files.	To generate and send to a printer, enter: `> DSAP /OUT $S.#print, NOWAIT/ $disk , DETAIL, SEPARATE`

See the *Guardian Disk and Tape Utilities Reference Manual* for the complete syntax, considerations, and examples of DSAP reports, and for error messages generated by DSAP.

Report Formats

When DSAP displays a report at a terminal, the report has an output width of 79 columns and is not formatted into pages. When DSAP sends a report to a spooler location, however, the report has an output width of 132 columns and a page length of 60 lines. The spooler report also has a header at the top of each page, which includes a page number. The SEPARATE option, in the example above, makes the information for each user start on a new page.

Examples

These examples show how to use DSAP to generate reports with information based on different specified criteria.

Generating a Brief Free Space Listing

To obtain a brief listing of free space on the disk $DATA1 with DSAP, enter:

```
> DSAP $DATA1, SHORT
```

DSAP displays a report such as this on your home terminal:

```
Disk Space Analysis Program -- T9543D20 - (01JUN93) -- 10/10/93   15:17:38
Tandem Computers Incorporated 1981, 1983, 1985-1993
Free Space Short Report
               -- Capacity (Mb) --   %     -- Free Extents --
Volume  (M)     Total      Free     Free    Count    Biggest
$DATA1   Y      1038      103.17      9       77       31.76
```

This report shows that the disk $DATA1 has a limited amount of free space.

Generating a Summary Report of Subvolumes

To generate a summary report of the subvolumes on the disk $DATA1, enter:

```
> DSAP $DATA1, BYSUBVOL
```

In addition to the information contained in the summary report shown in the previous example, DSAP returns a report such as this to your home terminal:

```
    .                        .       .     .      .       .     .       .
    .                        .       .     .      .       .     .       .
Summary of space use on $DATA1

       5,646 free pages in 319 extents (1.2%).

     428,645 allocated pages in 8,238 files in 33,153 extents (98.0%).

      38,032 unused pages in 4,978 files (8.6%).

Subvol Summary Report

 Subvolume Name                 Total Unused Dealloc Large  Min Age  Num
                       Files  Pages  Pages  Pages   File  Mod,Opn  Exp
FREE SPACE                      5646
DISK DIRECTORY                  2860
TEMPORARY FILES           35    2298    94     64     512    0,  0   35
ABCDE                      2     108     5      0     104   92, 48    2
EXCLNT                     4     946   194      0     556  188,188    4
FACETIOS                   1       4     3      0       4  188,146    1
    .                       .       .     .      .       .     .       .
    .                       .       .     .      .       .     .       .
    .                       .       .     .      .       .     .       .
XABCDEF                    1       4     3      0       4    5,  5    1
YELLOW                     1       4     3      0       4  564,564    1
ZEBRA                     17     674   153      0     180  502,  0   17
```

Generating a Summary Report of Specific Subvolumes

To run a DSAP report on the subvolumes owned by the user SALES.BONNIE (user ID 8,1) on the disk $DATA1, enter:

```
> DSAP $DATA1, BYSUBVOL, USER 8,1
```

DSAP displays a report such as this on your home terminal:

```
Disk Space Analysis Program -- T9543D20 - (01JUN93)  -- 7/20/93   10:27:22
Tandem Computers Incorporated 1981, 1983, 1985-1993

Summary of space use for SALES.BONNIE on $DATA1

         234 allocated pages in 8 files in 19 extents (0.0%).

          24 unused pages in 7 files (0.0%).

           0 deallocatable extent pages in 0 files (0.0%).

No SQL views.

Subvol Summary Report

 Subvolume Name                  Total Unused Dealloc Large  Min Age  Num
                        Files  Pages  Pages  Pages   File  Mod,Opn  Exp
BONNIEF                     6    162    23      0    140    6,  0    6
MEMOS                       2     72     1      0     64    0,  0    2
```

This report shows that SALES.BONNIE has eight files in two subvolumes.

Generating a Summary Report for a User

To generate a report for ACCTS.JUDY (user ID 8,76) on $DISK2, enter:

```
10> DSAP $DISK2, USER ACCTS.JUDY
```

or

```
10> DSAP $DISK2, USER 8,76
```

DSAP displays a report such as this on your home terminal:

```
Disk Space Analysis Program -- T9543D20 - (01JUN93) -- 3/2/94 14:35:20
Tandem Computers Incorporated 1981, 1983, 1985-1993

Summary of space use for ACCTS.JUDY on $DISK2

       1,958 allocated pages in 110 files in 512 extents (0.9%).

         253 unused pages in 72 files (0.1%).

           0 deallocatable extent pages in 0 files (0.0%).

No SQL views.
```

Generating a Summary Report of Subvolumes

To generate a report for ACCTS.JUDY on $DISK2, enter:

```
11> DSAP $DISK2, USER ACCTS.JUDY, BYSUBVOL
```

or

```
11> DSAP $DISK2, USER 8,76, BYSUBVOL
```

DSAP displays a report such as this on your home terminal:

```
Disk Space Analysis Program -- T9543D20 - (01JUN93) -- 3/2/94 14:35:20
Tandem Computers Incorporated 1981, 1983, 1985-1993

Summary of space use for ACCTS.JUDY on $DISK2

       1,962 allocated pages in 111 files in 513 extents (0.9%).

         255 unused pages in 72 files (0.1%).

           0 deallocatable extent pages in 0 files (0.0%).

No SQL views.

PAGE 1    DSAP -- ACCTS.JUDY -- Subvol Summary Report

 Subvolume Name                    Total Unused Dealloc Large  Min Age  Num
                         Files     Pages  Pages  Pages   File  Mod,Opn  Exp
TEMPORARY FILES             2        28     1      0      28    0,  0    2
ACCNTS                     74      1302   215      0     140    0,  0   73
JOBS                       29       562    30      0     228    1,  0   29
BUDGET                      1         4     0      0       4   41, 41    1
MISC89                      5        66     9      0      30    5,  0    5
```

Generating and Printing a User Detail Report

To generate a report for USERS.FRED (user ID 8,100) on $DISK2, and to send the report to the spooler location $S.#LP, enter:

```
12> DSAP /OUT $S.#LP/ $DISK2, USER USERS.FRED, DETAIL
```

or

```
12> DSAP /OUT $S.#LP/ $DISK2, USER 8,100, DETAIL
```

DSAP displays a report such as this on your home terminal, and sends it to the spooler:

```
Disk Space Analysis Program -- T9543D20 - (01JUN93) -- 3/2/94 14:35:20
Tandem Computers Incorporated 1981, 1983, 1985-1993

Summary of space use for USERS.FRED on $DISK2

       138 allocated pages in 9 files in 34 extents (0.9%).

         19 unused pages in 4 files (0.1%).

           0 deallocatable extent pages in 0 files (0.0%).

No SQL views.

PAGE 1    DSAP -- USERS.FRED -- User Detail Report

Selection Criteria: all files
   User                                  Total Unused Dealloc          Age
SQL
  Name/ID   Filename       Type Code     Pages  Pages  Pages  Exts
Mod,Opn Type
USERS.FRED
(8,100)  #0629                    440        28      1      0      7      6,
14
         #3043                                0      0      0      0      0,
0
         FRED.LTRS                101        12      0      0      6     55,
55
         FRED.STATUS              101         2      0      0      1      6,
6
         FRED.WORDLIST            101        36     10      0      3     55,
7
         FRED.TACLCSTM            101         4      2      0      1     42,
0
         FRED.TACLKEYS            101         2      0      0      1     33,
0
         FRED.TACLMACS            101        24      0      0     12      5,
0
         FRED.TEDPROFL          K   115        30      6      0      3     44,
0
```

Listing and Purging Old Disk Files

You can make more disk space available on your system by purging or compressing files and subvolumes.

The Disk Compression (DCOM) program is useful for compressing, or consolidating, disk space, and it can free significant amounts of disk space for users' files, applications, and processes.

You can make additional disk space available by analyzing the subvolumes and files contained on a disk and selectively purging old files. Also, when system users move to another system or leave their jobs, their files might be unneeded. Check with the managers of such users before you purge any files.

1. Determine whether old files exist on the disk.
   a. Log on as a super-group user (255,*n*).
   b. List all the subvolumes on a disk:

      ```
      > FUP SUBVOLS
      ```

A list of all existing subvolumes on the disk of your default subvolume is displayed. This list can help you determine the names of subvolumes that might need to be purged.

c. Generate a DSAP report showing all subvolumes owned by a given user:

```
> DSAP $disk, BYSUBVOL, USER group-id,user-id
```

d. Determine the ages of files in a given subvolume:

```
> VOLUME subvolume-name
> FILEINFO
```

A listing of all files contained within the given subvolume is displayed on your home terminal. The date of the most recent change to each file is shown in the column named “Last Modification.”

e. Consult with the user’s manager to determine whether you should purge any files. Your ability to purge files depends on file security settings; you might need to change file security if you are authorized to do so, or you might need to contact your operations manager or database administrator.

2. Purge the files or subvolumes you have determined are old.

- To purge a single file within a given subvolume, enter:

```
> PURGE subvolume.file
```

- To purge an entire subvolume, enter:

```
> FUP PURGE subvolume-name.*
```

Example

1. Generate a DSAP report that lists all the subvolumes on $DATA1 owned by the user SALES.BONNIE, the amount of disk space they occupy, and the number of days since the subvolumes have been modified and opened:

```
> DSAP $DATA1, BYSUBVOL, USER SALES.BONNIE
```

```
Disk Space Analysis Program -- T9543D20 - (01JUN93)  -- 7/23/93   10:28:29
Tandem Computers Incorporated 1981, 1983, 1985-1993

Summary of space use for SALES.BONNIE on $DATA1

         234 allocated pages in 8 files in 19 extents (0.0%).

          24 unused pages in 7 files (0.0%).

           0 deallocatable extent pages in 0 files (0.0%).

No SQL views.

Subvol Summary Report

 Subvolume Name                  Total Unused Dealloc Large  Min Age  Num
                          Files  Pages  Pages  Pages   File  Mod,Opn  Exp
BONN                          6    162     23      0    140  365,365    6
MEMOS                         2     72      1      0     64  365,365    2
```

This report shows that SALES.BONNIE has only eight files in two subvolumes named BONN and MEMOS. The columns “Min Mod” and “Age Opn” both show that these subvolumes have not been modified or opened in 365 days. When the number of days since these files have been modified or opened exceeds 999, the DSAP program reports 1K+ in the “Min Mod” and “Age Opn” columns.

2. Determine the date that changes were last made to the files in the subvolume BONN:

```
> VOLUME BONN
> FILEINFO
```

A report such as this is displayed on your home terminal:

```
$DATA1.BONN

              Code           EOF Last Modification   Owner   RWEP   PExt  SExt
LOG            101          2092 13-Jul-91 14:41:08   8,001 "NUNU"     6     6
MISC           101        267022 10-Jul-91 14:04:51   8,001 "NUNU"    14    14
MYMACS         101          5670 13-May-91 13:27:57   8,001 "NUNU"     2     2
SCFCSTM        101           144  7-Jul-91 13:33:16   8,001 "NUNU"     2     2
SYSCHK         101           672 14-Jul-91 11:22:05   8,001 "NUNU"     6     6
TACLCSTM       101            80  9-Jul-90 16:59:25   8,001 "NUNU"     4    16
```

This report shows that no files have been changed since July, 1991.

3. Purge the file MISC:

```
> PURGE BONN.MISC
```

4. Purge the subvolume BONN:

```
> FUP PURGE BONN.*
```

FUP responds with:

```
 DO YOU WISH TO PURGE THE ENTIRE SUBVOLUME $DATA1.BONN
 ( Y[ES], N[ONE], S[ELECT], F[ILES] )?
```

Enter Y to complete this operation.

Monitoring and Altering Swap Files

When all physical memory has been allocated and more memory is needed, data that is not currently in use is stored on disk:

- Pages of memory are “swapped,” or copied, to disk when there is a shortage of available physical memory.
- The memory pages are swapped back to physical memory when the data is accessed.
- When swapped to disk, the data is stored in “swap files.”
- The NonStop™ Kernel opens one or more swap files for each processor and manages the files for all the processes that need them.

A kernel-managed swap file is only opened once and is then available to all the processes running on the processor. Conventional swap files, which are defined by the calling process rather than the system, must be opened and closed by the system monitor on each process creation and deletion.

Kernel-managed swap files, which you can control using the Kernel-Managed Swap Facility (KMSF), offer four main benefits over conventional swap files:

- Kernel-managed swap files speed up process creation and deletion.
- KMSF uses much less disk space than conventional swap files when backing the large, sparsely populated address space used by shared run-time libraries (SRLs).
- KMSF reduces the time required to resize a segment. The KMSF implementation of segment resizing requires neither I/O nor dispatches of other processes.
- By centralizing swap files, KMSF makes it easier to manage swap space for all processes on the system.

For more information, see the *Kernel-Managed Swap Facility (KMSF) Manual.*

How Kernel-Managed Swap Space Works

When a processor is loaded, KMSF reads the kernel-managed swap configuration file and opens any swap files configured for that processor. If no configuration file or swap files are found, KMSF attempts to create a default swap file. Each swap file is assigned to only one processor. There is no limit to the number of swap files that you can have for each processor—you are limited only by the amount of disk space available.

KMSF receives requests for swap space (which can also be thought of as virtual memory) from the NonStop™ Kernel and returns swap-space reservations to the NonStop™ Kernel. A reservation is the agreement to provide up to a stated amount of space as needed; the space is not allocated all at once, but is allocated as it is needed. The initial reservation is the amount of swap space requested at creation of a process.

The NonStop™ Kernel swaps to the kernel-managed swap files as needed. As a process’s need for swap space grows, KMSF increases the reservation. Additional swap space might be given from a different swap file than that used for the original

reservation. When a process no longer needs swap space, the NonStop™ Kernel returns it to KMSF.

Components of Kernel-Managed Swapping

Kernel-managed swapping is implemented through:

- The Kernel-Managed Swap Facility (KMSF), which reads the kernel-managed swap configuration file, controls the swap files and provides, resizes, and receives back swap space reservations and swap space allocations.
- The NonStop™ Kernel memory manager, which acquires, requests resizes of, and returns swap space reservations, and requests and returns swap space allocations.
- NSKCOM, the NonStop™ Kernel utility used to configure and manage kernel-managed swap files.
- ZSYSCFG, the configuration file specifying the names and usage of kernel-managed swap files.
- The kernel-managed swap files.

Kernel-Managed Swap Process Flow

When a process is created, the NonStop™ Kernel memory manager requests a single swap space reservation for the process from KMSF. If there is a shortage of available physical memory, the memory manager finds a page suitable for swapping, such as a page belonging to an inactive process. If the chosen page contains dynamic data (rather than code or read-only data), its contents must be saved on disk. The memory manager requests swap space from KMSF, which finds and allocates swap space against the reservation.

How Kernel-Managed Swap Files Affect You

KMSF, which centralizes the control of swap space under the NonStop™ Kernel, has implications for users, operations personnel, and application developers.

KMSF and Users

When a user starts a process, KMSF provides the swap space that the process requires at process creation. User control over certain aspects of swap files is limited, because kernel-managed swap files provide swap space for multiple processes:

- In most cases, users cannot decide where data will be swapped. The user can specify the location of the swap space for extended data segments only. Otherwise, data is swapped to whatever kernel-managed swap file or files are available for the processor in which the process is running.
- Only super-group users (255,*n*) can resize, add, or delete kernel-managed swap files.

KMSF and Operations

KMSF affects installation and configuration. For kernel-managed swap files, system administrators must plan for and configure:

- How much swap space is needed
- Where to place swap files
- What guidelines they need to create for operations staff on monitoring and altering swap files

KMSF also affects routine operations. Operations staff should monitor KMSF and operator messages to spot potential problems and dynamically add swap space as needed. Event Management Service (EMS) messages are generated to alert staff to swap files that have reached a configured threshold and to changes in KMSF configuration.

KMSF does not use more disk space than conventional swap files because the size and number of temporary files that were formerly used for swapping decline as usage of KMSF increases.

KMSF and Applications

Under KMSF, the memory needs of processes, including the main stack, globals, heap space, extended data segments, and shared run-time library (SRL) instance data, are copied to a common swap file or files managed by the NonStop™ Kernel.

You do not need to recode applications for KMSF.

Using NSKCOM to Monitor and Alter Swap Files

Use the NSKCOM utility of KMSF to monitor your kernel-managed swap files to ensure that you have adequate swap space available. If adequate swap space is not available, it might prevent process creation and cause processes to fail. You can also use NSKCOM to configure and manage swap files if you need to make any changes.

Monitoring Swap Files

To monitor kernel-managed swap files, you can use NSKCOM to review the status of the swap files and the swap file statistics for each processor.

1. Run NSKCOM.

```
> NSKCOM
```

When NSKCOM starts, it automatically displays your current configured swap files. The NSKCOM opening banner does not display default swap files.

```
$SYSTEM SYS66 35> NSKCOM
NSKCOM - T9050D42 BASE (01JULY96)
Copyright [c] 1995, Tandem Computers Incorporated
$SYSTEM.SYSTEM.ZSYSCFG
KMS.SWAPFILE = 0 $SWAP0.SYSSWAP.SWAP00 THRESHOLD 80
KMS.SWAPFILE = 1 $DATA1.SYSSWAP.SWAP01
KMS.SWAPFILE = 2 $SWAP.SYSSWAP.SWAP02
KMS.SWAPFILE = 3 $SWAP3.SYSSWAP.SWAP03
NSK-
```

In the above example:

- The numbers 0, 1, 2, and 3 represent the CPUs.
- The threshold indicates the usage threshold at which an EMS message is generated. If blank, the file has the default threshold of 85 percent.
- NSK- is the NSKCOM prompt.

2. Determine the size and usage of your kernel-managed swap files.

```
> NSK- STATUS SWAPFILE
```

This command displays a swap file's total number of processor pages, as well as the pages that are allocated and available, as follows:

```
NSK-STATUS SWAPFILE *

Status of $SWAP.SYSSWAP.SWAP00
CPU 0
CPU Pages: Total 4096   Reserved 88      Available 4008
Peak CPU Pages ever reserved 92
Threshold 3481 CPU pages

Status of $SWAP.SYSSWAP.SWAP01
CPU 1
CPU Pages: Total 4096   Reserved 68      Available 4028
Peak CPU Pages ever reserved 80
Threshold 3481 CPU pages
...
```

3. Display statistics for each processor, using the STATUS KMSF command.

```
> NSK- STATUS KMSF
```

A report such as the following is sent to your home terminal:

```
NSK-STATUS KMSF

KMSF statistics from CPU 0
Total swap space 560 MB
Historical data:
Reservations:   Creates 934 (failures 0)        Releases 911 (failures 0)
                Resizes 6 (failures 0)
Actual use:     Allocates 19 (failures 0)       Frees 3 (failures 0)
Reserved CPU Pages 60 (for 23 Reservations)     Available pages 4100

KMSF statistics from CPU 1
Total swap space 560 MB
Reserved CPU Pages 412 (for 35 Reservations)    Available pages 4110
Historical data:
Reservations:   Creates 46 (failures 0)         Releases 28 (failures 0)
                Resizes 16 (failures 0)
Actual use:     Allocates 0 (failures 0)        Frees 0 (failures 0)
NSK-
```

Total swap space	Total number of megabytes (MB) that are currently allocated in kernel-managed swap files for the processor.
Reserved CPU Pages	Total memory pages that are currently allocated for the processor.
Available pages	Total number of memory pages currently available in swap files for the processor.
Creates	Total number of reservations made for process creations since the processor loaded.
Releases	Total number of reservations released since the processor loaded.
Resizes	Total number of reservations that have been resized since the processor loaded.
Allocates	Total number of allocations that have been made in the swap files.
Frees	Total number of allocations that have been freed.

- The difference between the creates and releases equals the current number of reservations.
- The difference between allocates and frees is the actual number of pages currently being used for storing swap data.
- This actual current usage can be at most equal to the number of Reserved CPU Pages at any given time.

For more information about NSKCOM, see the *Kernel-Managed Swap Facility (KMSF) Manual*.

Solving Common Disk Problems

Table 9-2. Common Disk Problems

Problem	Symptoms	Solution
Defective tracks or sectors exist.	SCF INFO DISK, BAD output shows unspared defective sectors.	Spare defective sectors (use the SCF CONTROL DISK, SPARE command) if you have the authorization to do so.
	SCF INFO DISK, LOG output shows a large number of spared defective sectors.	Back up files to tape, reformat disk, and restore files to disk.
Performance problems are occurring because of path switches.	Users report poor application performance or an SCF STATUS DISK command shows a nonstandard configuration.	Use the SCF PRIMARY DISK command to reverse primary and backup disk processes.
Disk free-space fragmentation has occurred.	Error 43 occurs (unable to obtain disk space for file extent).	Use the DCOM program to consolidate disk space usage.
A disk is full.	Error 43 occurs.	Ask users to purge files.
	An application might go down.	Identify large, old, and little-used files with DSAP. Back up such files or purge them if you have the authority. (See Listing and Purging Old Disk Files on page 9-20.)
	An Object Monitoring Facility (OMF) warning can occur (if threshold has been set).	If you are authorized, move files to another disk. If you are not authorized, notify your management.
One disk in a mirrored pair is down.	An event message is generated, but the application continues to run.	Use the SCF START DISK command. If this does not bring the disk back up, notify your management; the disk might need to be repaired or replaced.
An unmirrored disk is down or both disks in a mirrored pair are down.	Users report access problems, an application goes down, and event messages can be generated.	Escalate the problem. (Database recovery or disk repair might be required.)

10 Using Labeled Tapes

Tapes and the tape drives that run them are an integral part of the operations environment, because most companies rely on tapes to back up and store critical data.

A labeled tape contains a record at the beginning of the tape that identifies the tape volume and the files on the tape. If labeled-tape processing is enabled on your system, you can use labeled tapes with programs such as BACKUP, RESTORE, and FUP. Compaq supports both ANSI and IBM labeled-tape formats.

You must use a TAPE DEFINE to access a file on a labeled tape. A TAPE DEFINE is a named set of attributes and values that you use to specify information about a tape file such as the volume ID, tape density, and operator mount messages.

This section explains how to use the MEDIACOM program to perform routine operations relating to the tapes and tape drives on your system:

For more information about using MEDIACOM, see the *DSM/Tape Catalog Operator Interface (MEDIACOM) Manual*.

MEDIACOM replaces the TAPECOM utility. If you have not yet switched from TAPECOM and need more information about it, see the *Guardian Disk and Tape Utilities Reference Manual.*

How Labeled-Tape Processing Works

When labeled-tape processing is enabled for your Compaq *NonStop*™ Kernel system during system generation, to perform a typical mounting of a labeled-tape:

1. A user submitting a job to run sends a message to the operator console requesting:
   - A specific tape be mounted on any tape drive or on a specific tape drive
   - Permission to use a specific tape drive, if needed
2. The operator mounts the requested tape.

   The system uses automatic volume recognition (AVR) to check the labels on the mounted tape to ensure that the requested tape has been mounted.

   If a user requests that a specific tape be mounted, and the operator mounts the wrong tape, the tape remains online and a message similar to this is displayed:

```
UNREQUESTED ANSI TAPE AMBER1 ONLINE ON $TAPE1
```

   A job does not start until the system determines that the requested tape is mounted.

The MEDIACOM Interface

Some MEDIACOM commands require that you log on as a super-group user (255,*n*), but in general MEDIACOM lets you:

- Configure DSM/TC
- Label and catalog new tapes
- Handle tape mount requests
- Retrieve information from DSM/TC catalog entries
- Manage the use of uncataloged tapes

To run MEDIACOM, enter:

```
> MEDIACOM
```

You can use the MEDIACOM commands and options in Table 10-1 in an environment other than DSM/TC. There are other MEDIACOM commands and options not described in this guide that you can use only with systems that have a DSM/TC database installed.

When you enter HELP at the MEDIACOM prompt, all commands are displayed. Each MEDIACOM command has a HELP option that lists full syntax and options. For example, when you enter

```
MC> HELP STATUS TAPEDRIVE
```

MEDIACOM returns:

```
STATUS TAPEDRIVE [ \node.] $tape ] [ , BRIEF | DETAIL ]
```

Table 10-1. MEDIACOM Commands (page 1 of 2)

Command	Description
!	Retrieves and executes a previously entered command
?	Retrieves a previously entered command
ACCEPT TAPEMOUNT	Permits use of a tape drive for no label processing or bypass label processing
ADD POOL	Creates a new pool in a volume catalog
ADD TAPEFILE	Catalogs an existing tape file
ADD TAPELABEL	Creates or changes a tape label
ADD TAPEVOLUME	Catalogs a tape in a pool and volume catalog
ALTER CONSOLE	Designates a device as the tape console
ALTER DISKFILE	Changes the status of a disk file entry to valid or invalid
ALTER FILECAT	Changes the operating system security level or file catalog owner
ALTER MEDIADEFS	Sets or changes the system default values
ALTER POOL	Changes one or more attributes of an existing pool
ALTER TAPEDRIVE	Sets or changes the tape drive mode for no label or bypass label processing; also assigns tape drive name to ACS tape drive
ALTER TAPEFILE	Changes the tape file entry retention period or expiration date, and the tape file entry status that represents a file in a multifile tape set
ALTER TAPEMOUNT	Requests a labeled tape be mounted other than the currently requested one
ALTER TAPEVOLUME	Changes the status of a tape in the volume catalog
ALTER VOLCAT	Changes operating system security level or volume catalog owner
CREATE FILECAT	Creates a file catalog; assigns it a logical name, owner, and operating system security level
CREATE VOLCAT	Creates a volume catalog; assigns it a logical name, owner, and operating system security level
DELETE POOL	Removes a pool once it is empty
DELETE TAPEFILE	Removes an unexpired tape file entry
DELETE TAPELABEL	Removes a standard label from a labeled tape
DELETE TAPEVOLUME	Removes a tape entry from a volume catalog if the tape's status is bad, released, or scratch
DROP FILECAT	Removes a file catalog once it is empty
DROP VOLCAT	Removes a volume catalog once it is empty
ENV	Displays the current session default values
EXIT	Ends the session; returns control to the command interpreter

Table 10-1. MEDIACOM Commands (page 2 of 2)

Command	Description
EXPIRE TAPEFILE	Runs the clean-up process to remove expired entries permanently
FC	Retrieves, edits, and executes a previously entered command
FILECAT	Defines a session default name for a file catalog
HELP	Displays the names and syntax of MEDIACOM commands
HISTORY	Displays or removes commands saved in the history buffer
INFO CONSOLE	Retrieves the device name of the tape console
INFO DISKFILE	Retrieves information about valid and invalid disk file entries
INFO FILECAT	Retrieves a file catalog's logical and physical names, owner, and security level
INFO MEDIADEFS	Retrieves information about the DSM/TC environment
INFO POOL	Retrieves attributes of a pool
INFO TAPEDRIVE	Displays current setting of a tape drive for the processing mode
INFO TAPEFILE	Retrieves information about a tape file entry
INFO TAPELABEL	Identifies and displays a tape's label
INFO TAPEVOLUME	Retrieves attributes of a tape and any files written to it
INFO VOLCAT	Retrieves a volume catalog's logical and physical names, owner, and security level
LABEL TAPEVOLUME	Creates or changes a tape label for a cataloged tape
OBEY	Executes commands stored in a file
OPEN SERVER	Creates a MEDIACOM server dedicated to your session
PAUSE	Lets another process control the home terminal during the session
POOL	Defines a session default name for a pool
RECOVER DISKFILE	Copies a disk file from a backup tape to disk
REJECT TAPEMOUNT	Cancels a request to use a tape drive or tape
STATUS SERVER	Displays MEDIACOM server information for your session
STATUS TAPEDRIVE	Retrieves information about the status of a tape drive
STATUS TAPEMOUNT	Retrieves information about the outstanding tape mount requests on a node
TAPEDRIVE	Defines a session default name for a tape drive
UNLOAD	Defines session default for whether a tape is left online after a command finishes
VOLCAT	Defines a session default name for a volume catalog
VOLUME	Defines session default for a node, volume, or subvolume name of a disk file ID

Tape Processing Modes

If you use labeled-tape processing, you can open a tape file in one of three tape processing modes:

Standard Label Processing (LP) Mode

1. An application generates a mount request at the operator console for a specific labeled tape.
2. The operator can:
   - Mount the requested labeled tape, or
   - Reject the request

Unlabeled or No Label Processing (NL) Mode (Default)

1. Mount requests are not generated for unlabeled tapes. (Labeled tapes are rejected.)
2. The system processes the open according to the state of NLCHECK for that drive:
   - If NLCHECK is on, a super-group user (user ID 255, *n*) must give permission for the open using a TAPECOM ACCEPT command for the tape drive.
   - If NLCHECK is off, permission is not required.

Bypass Label Processing (BLP) Mode

1. An application generates a request at the operator console for a specific tape drive.
2. The system processes the open according to the state of BLPCHECK for that drive:
   - If BLPCHECK is on, a super-group user (user ID 255, *n*) must give permission for the open using a TAPECOM ACCEPT command for the drive.
   - If BLPCHECK is off, permission is not required.
3. If the operator accepts the request:
   a. The system unloads any mounted labeled tape.
   b. The operator mounts the tape to be used.
   c. The operator must ensure that the correct tape is mounted since the system does not check the tape.

Using Labeled Tapes in LP Mode

1. Create a TAPE DEFINE for each file you want to access on the labeled tape.

   To specify LP mode, you must include LABELS ANSI, LABELS IBM, LABELS IBMBACKUP, or LABELS BACKUP in the TAPE DEFINE.

   This example shows a TAPE DEFINE named =TAPE-JOB for LP mode:

```
10> ADD DEFINE =TAPE-JOB, CLASS TAPE, LABELS ANSI, &
10> &VOLUME TV0005, DEVICE $TD2, USE OUT, &
10> &MOUNTMSG "Low-priority. Runs 30 min. Thanks, Tom"
11>
```

2. Mount the tape, or deliver the tape to the operator with instructions that include:
   - The tape identification (volume ID or serial number)
   - The open mode (LP)
   - The tape label type (ANSI, IBM, or BACKUP)
   - The name of the tape drive your program will use (if you included a DEVICE attribute in your DEFINE)
   - The time you will run the application that requires the tape
3. Run your application, specifying the DEFINE that describes the tape file you want. If your application lets you specify the TAPE DEFINE on the TACL command line, you must enter the correct name of the TAPE DEFINE.

   For example, this application copies the disk file ACCOUNTS to a labeled tape using the DEFINE =TAPE-JOB:

```
11> RUN TAPECOPY / OUT =TAPE-JOB, NOWAIT / ACCOUNTS
```

   The labeled-tape server displays this mount message at the operator console:

```
$ZSVR: 0012 MOUNT TV0005 ON $TD2 WITH RING
"Low-priority. Runs 30 min. Thanks, Tom"
```

   This mount message includes:
   - The name of the labeled-tape server process $ZSVR and a message sequence number (0012) identifying the request
   - The word MOUNT to indicate the operation requested
   - The volume serial number (TV0005) of the tape to be mounted
   - The phrase WITH RING to indicate that the tape should be write-enabled
   - The comment you included in the DEFINE MOUNTMSG attribute
4. You, or an operator, should mount the tape TV0005. The system uses automatic volume recognition to identify the tape so your application can continue processing.

Using Unlabeled Tapes in NL Mode

NL mode is the default mode for tape processing. You can select NL mode by:

- Not using a TAPE DEFINE, or
- Specifying LABELS OMITTED in a TAPE DEFINE.

Using an Unlabeled Tape Without a TAPE DEFINE

1. Mount the unlabeled tape, or deliver it to the operator with instructions that include:

   - The open mode (NL)
   - The name of the tape drive your application will use
   - The time you will run your application that needs the tape

   If your application opens a specific tape drive by name, such as $TAPE1, or if your application lets you specify a tape drive on the TACL command line, you must use the correct name of the tape drive.

   Processing continues according to the value of NLCHECK (which you can check using the TAPECOM SHOW NLCHECK command):

   - If NLCHECK is on, the labeled-tape server $ZSVR displays this drive request at the operator console:

```
$ZSVR: 0125 REQUEST TO USE $TAPE2 UNLABELED
```

     This request includes:

     - The name of the labeled-tape server process ($ZSVR) and a message sequence number (0125) identifying the request
     - The phrase REQUEST TO USE indicating that this is a request to use a tape drive
     - The name of the requested tape drive ($TAPE2)
     - The phrase UNLABELED indicating NL mode

     a. You, or an operator, should mount an unlabeled tape on tape drive $TAPE2.

     b. Issue a TAPECOM ACCEPT command to let your application use this tape. (A TAPECOM REJECT command causes your open request to fail with file-system error 194.)

   - If NLCHECK is off, processing continues depending on the application you are running. If your application allows time for you or an operator to mount a tape, mount the tape on the specific tape drive ($TAPE2) when your application requests it. Your application can then continue processing.

If your application does not allow time to mount the tape, the results can be unpredictable. In this case, mount the tape before the application needs it.

Note. If a standard labeled tape is mounted on a tape drive that is opened in NL mode, the system rejects the tape and displays an error message.

Using an Unlabeled Tape With a TAPE DEFINE

1. Create a TAPE DEFINE for each tape file.

   For NL mode, specify LABELS OMITTED and the name of the tape drive you want to use for DEVICE.

   This example shows a TAPE DEFINE named =NOLABEL_RUN for NL mode:

```
20> ADD DEFINE =NOLABEL_RUN, CLASS TAPE, &
20> &LABELS OMITTED, DEVICE $TD1, &
20> &MOUNTMSG "Employee subvolume copy. Need by 3:00 PM."
```

2. Mount the tape, or deliver the tape to the operator with instructions that include:
   - The open mode (NL)
   - The name of the tape drive your application will use
   - The time you will run the application that needs the tape
3. Run your application.

   If you run an application, make sure that your application specifies the TAPE DEFINE that describes the tape file you want. If your application lets you specify the TAPE DEFINE on the TACL command line, you must enter the correct name of the TAPE DEFINE.
4. Mount an unlabeled tape depending on the value of NLCHECK, as described in Using an Unlabeled Tape Without a TAPE DEFINE on page 10-7.

Bypassing Label Protection in BLP Mode

In BLP mode, the system does not check the label of the tape mounted on the drive. If you issue a TAPECOM ACCEPT command, there is no protection against overwriting data on the tape.

To open a tape file in BLP mode:

1. Create a TAPE DEFINE for the tape.

   For BLP mode, specify LABELS BYPASS and the name of the tape drive for DEVICE.

   This example shows a TAPE DEFINE named =BYPASS-JOB for BLP mode:

```
30> ADD DEFINE =BYPASS-JOB, CLASS TAPE, DEVICE $DR2, &
30> &MOUNTMSG "HIGH-PRIORITY! Call when finished"
```

2. Mount the tape, or deliver the tape to the operator with instructions that include:
   - The open mode (BLP)
   - The name of the tape drive your application will use
   - The time you will run the application that requires the tape
3. Run your application.

   If you run an application, make sure that your application specifies the TAPE DEFINE that describes the tape file you want. If your application allows you to specify the TAPE DEFINE on the TACL command line, you must enter the correct name of the TAPE DEFINE.

   If BLPCHECK is on, the labeled-tape server displays the following request at the operator console:

```
$ZSVR: 0341 REQUEST TO USE $TD2 WITH NO LABEL PROTECTION
"HIGH-PRIORITY! Call when finished"
```

4. You, or an operator, should mount the tape on the specified drive.
5. Issue a TAPECOM ACCEPT command.

   The system does not check whether the mounted tape is the correct one before your application continues processing. A TAPECOM REJECT command causes the open request to fail with file-system error 194.

TAPE DEFINE Attributes

Table 10-2. TAPE DEFINE Attributes (page 1 of 2)

Attribute	Function
BLOCKLEN	Data block size in bytes for a tape file
DENSITY	Tape density in bits per inch (bpi)
DEVICE	Tape drive on which the tape file is to be mounted
EBCDIC	Specifies whether data is translated when processing an IBM tape
EXPIRATION	Expiration date for the tape file
FILEID	Tape file name
FILESECT	Order of the volume (or section of file) within a multivolume file being created at the same time
FILESEQ	Relative position of this tape file in a multifile volume set
GEN	Indicates that this file is part of a generation group
LABELS	Type of tape; for labeled tapes, the label processing mode
OWNER	Owner ID in the VOL1 label for IBM labeled tapes only
MOUNTMSG	Mount message to be displayed with the system mount message or tape drive use request

Table 10-2. TAPE DEFINE Attributes (page 2 of 2)

Attribute	Function
RECFORM	Tape record format
RECLEN	Record length for the tape file
REELS	Number of volumes (reels or cartridges) in a multivolume file
RETENTION	Retention period for the tape file
SYSTEM	Name of the system that controls the tape drive
TAPEMODE	Specifies the operating mode for a cartridge tape drive
USE	Specifies how the tape file is to be used (as input or output)
VERSION	Version within one generation
VOLUME	Six-byte tape volume ID identifying the tape, or SCRATCH to indicate that any scratch tape is acceptable

Common Labeled Tape Activities

There are several basic tasks you might need to perform as part of your labeled tape activities, including:

Checking the Status of Tape Drives

You can list the tape drives on your system and determine their status by using either MEDIACOM or SCF.

Checking Status Using MEDIACOM

To check the status of all tape drives on your system with MEDIACOM, enter:

```
> MEDIACOM STATUS TAPEDRIVE
```

A listing similar to this is sent to your home terminal:

```
MEDIACOM - T6028D20 (01JUN93)
Copyright Tandem Computers Incorporated 1993
                Drive    Tape    Tape     Label  Open
Tape Drive      Status   Name    Status   Type   Mode   Process Name
----------      ------   ----    ------   -----  ----   ------------
$TAPE1          INUSE    TT0046  ASSIGNED ANSI   LP     \SKY.$BURT
$TAPE2          INUSE    TT0047  ASSIGNED ANSI   LP     \SKY.$SID
```

The fields in this display are explained in the *DSM/Tape Catalog Operator Interface (MEDIACOM) Manual*.

Example

To obtain status information about the tape drive $TAPE1 by using MEDIACOM, enter:

```
> MEDIACOM STATUS TAPEDRIVE $TAPE1
```

A listing such as this is sent to your home terminal:

```
MEDIACOM - T6028D20 (01JUN93)
Copyright Tandem Computers Incorporated 1993
                  Drive    Tape    Tape     Label  Open
Tape Drive        Status   Name    Status   Type   Mode   Process Name
----------        ------   ----    ------   -----  ----   ------------
$TAPE1            INUSE    TT0046  ASSIGNED ANSI   LP     \SKY.$BURT
```

This listing shows that $TAPE1 is in use. For more information about MEDIACOM, the listings it generates, and the tasks it enables you to perform, see the *DSM/Tape Catalog Operator Interface (MEDIACOM) Manual*.

Checking Status Using SCF

To check the status of all tape drives on your system with SCF, enter:

```
> SCF INFO TAPE $TAPE0
```

A listing similar to this is sent to your home terminal:

```
Storage - Info TAPE $TAPE0
SubType  Density   NumOpeners  RecSize  TapeMode  Compression
9        38000     6           8192     STREAM    ON
```

This report shows that $TAPE0:

- Is subtype 9 (either a 5190 or 5194 tape drive)
- TapeMode is STREAM (tape continuously moves while reading or writing)
- Has compression ON (increases the capacity of the tape to store data)

SubType	The subtype number of the tape drive.
Density	The tape density for the tape drive.
NumOpeners	The maximum number of concurrent open files allowed for the tape drive.
RecSize	The configured record size for the tape drive.
TapeMode	For cartridge tape drives, the mode in which it operates (STREAM or STARTSTOP). For other tape drives, this field is not applicable (N/A).
Compression	For cartridge tape drives, whether it is configured for compression (ON) or not (OFF). For other tape drives, this field is not applicable (N/A).

See Table 19-2, SCF Object States, on page 19-6 for more information about the possible states of tape drives and other devices. For complete information about SCF, see the *SCF Reference Manual for G-Series Releases*.

Example

To obtain status information about the tape drive $TAPE1 by using SCF, enter:

```
> SCF INFO TAPE $TAPE1
```

A listing such as this is sent to your home terminal:

```
Storage - Info TAPE $TAPE1
SubType  Density   NumOpeners  RecSize  TapeMode  Compression
9        38000     6           8192     STREAM    ON
```

Setting a Default Tape Drive

One of the first steps for setting up your labeled-tape environment is to set a default tape drive for MEDIACOM sessions.

Many MEDIACOM commands require that you specify a tape drive. Use the MEDIACOM TAPEDRIVE command to select a default tape drive for the current session.

To set a default tape drive with MEDIACOM, enter:

```
> MEDIACOM
MC> TAPEDRIVE $tape
```

$*tape* is the name of the tape drive you want to specify as the default device.

The drive you specify remains in effect until you enter another TAPEDRIVE command or you end the MEDIACOM session.

Guidelines

- When writing to a tape that requires a certain recording density, make sure that the specified drive supports that density.
- MEDIACOM does not support 7-track tape drives.
- The named device can be a network name because MEDIACOM operates in distributed systems environments.
- Any subsequent device parameter in a MEDIACOM command does not change this default assignment but does specify the device to be used for that command.
- The effect of the TAPEDRIVE command is valid only for the current session of MEDIACOM.

Example

1. To make $TAPE1 the default drive with MEDIACOM, enter:

   ```
   MC> TAPEDRIVE $TAPE1
   ```

2. Check that $TAPE1 is designated as the default tape drive for the current MEDIACOM session:

   ```
   MC> ENV
   ```

   A listing such as this is sent to your home terminal:

```
Tape Drive              \SAGE.$TAPE1
File Catalog            \SAGE.FILE_CATALOG_SAGE
Volume Catalog          \SAGE.MASTER_TAPE_CATALOG
Pool Name               TAPES_FOR_SAGE
Volume                  \SAGE.$OPRVOL.OPR
Unload                  ON
```

Taking Down and Bringing Up a Tape Drive

You must bring down a tape drive when it requires maintenance, if it is going to be replaced, or sometimes for other reasons. After taking a drive down, you need to bring it back up before you can use it again.

Taking Down a Tape Drive

1. Log on as a super-group user (255,*n*).
2. Stop the tape drive:

   ```
   > SCF STOP TAPE $tape
   ```

Example

1. Log on as a super-group user (255,*n*).
2. Enter:

   ```
   > SCF STOP TAPE $TAPE1
   ```

Bringing Up a Tape Drive

1. Log on as a super-group user (255,*n*).
2. Start the tape drive from a stopped state:

   ```
   > SCF START TAPE $tape
   ```

Example

1. Log on as a super-group user (255,*n*).
2. Enter:

   ```
   > SCF START TAPE $TAPE1
   ```

3. Check to make sure that $TAPE1 is actually up:

```
> SCF STATUS TAPE $TAPE1, DETAIL
```

```
STORAGE - Detailed Status TAPE \ALM171.$TAPE1
BackupPID................... 1, 30
LDev........................ 20
PrimaryPID.................. 0, 30
State....................... STARTED
SubState....................
```

Handling Labeled Tape Messages and Requests

System operators must monitor for labeled-tape messages so they will know when errors occur or requests for tape handling are sent. Labeled-tape requests are sent to operators asking them to perform various tasks such as mounting or taking down a labeled-tape.

Monitoring Labeled-Tape Messages

All messages generated by labeled-tape operations are displayed at the operator console and any other enabled terminal. These messages are generated either from the tape processes or from the $ZSVR (labeled-tape server process) process.

$ZSVR is the central control process for all tape processes that support labeled tapes in the system. Every tape drive in the system has a tape process associated with it. $ZSVR monitors the status of each tape drive—whether it is free or in use, whether a tape is currently mounted, and what the volume serial number is (if a standard labeled tape is mounted). $ZSVR acts as the message interface between all tape processes and the operator console. When a job wants to open a standard labeled tape or a tape drive, $ZSVR sends the request to the operator console. $ZSVR also sends error messages and status messages to the operator console.

Tape processes read the labels on mounted tapes and send the information to $ZSVR. Tape processes also perform the actual reading and writing of tapes. If a tape process encounters any problems while reading the label information on a mounted tape, it sends an error message or warning message to the operator console.

Directing Labeled-Tape Messages to a Second Console

You can direct all labeled-tape processing messages to a second console in addition to the operator console by using the MEDIACOM ALTER CONSOLE command. This lets you separate labeled-tape messages from other console messages. If you enable a second console, labeled-tape messages will also continue to appear on the operator console.

The MEDIACOM ALTER CONSOLE command displays outstanding mount, drive-usage, and unknown tape messages and then puts the current MEDIACOM process in console mode.

To direct labeled-tape messages to a console other than the operator console with MEDIACOM, log on as a super-group user (255,*n*) and enter at that terminal:

```
> MEDIACOM ALTER CONSOLE, DEVICE terminal-name
```

This command also displays outstanding tape mount and drive-usage requests and puts the specified terminal in console mode. If you do not specify a device, the current terminal is used for tape mount and drive-usage requests.

Only one terminal in a system can be in console mode at a time. If you enter the CONSOLE or ALTER CONSOLE command when another terminal is already in console mode, an error message is displayed.

Use the TACL STATUS command to list the name of the terminal running the process (*cpu*,*pin*):

```
> STATUS cpu,pin
```

To terminate console mode with MEDIACOM, enter the ALTER CONSOLE, RESET DEVICE command at the terminal you have designated as the console.

Example

To use MEDIACOM to view the messages sent from $ZSVR on your terminal, log on as a super-group user (255,*n*) and enter:

```
> MEDIACOM
```

```
MEDIACOM - T6028D20 (01JUN93)
Copyright Tandem Computers Incorporated 1993
```

```
MC> ALTER CONSOLE
```

Following are some examples of labeled-tape processing messages:

```
$ZSVR : STATUS 1505 -        UNREQUESTED NL TAPE ONLINE ON $TAPE
$ZSVR : STATUS 1504 -        TAPE OPENED ON $TAPE
$ZSVR : STATUS 1512 -        TAPE DISMOUNTED FROM DRIVE $TAPE
$ZSVR : STATUS 1504 -        TAPE OPENED ON $TAPE
$ZSVR : STATUS 1512 -        TAPE DISMOUNTED FROM DRIVE $TAPE
$ZSVR : STATUS 1504 -        TAPE OPENED ON $TAPE1
$ZSVR : STATUS 1512 -        TAPE DISMOUNTED FROM DRIVE $TAPE1
```

Responding to Messages and Requests

Labeled-tape operation messages fall into these categories:

- $ZSVR error and status messages
- Tape process error and warning messages
- Labeled tape and scratch tape mount requests
- Tape drive-usage requests
- Operator attention requests

All labeled-tape messages that have a sequence number require some action from you. Sequence numbers are used in the MEDIACOM ACCEPT TAPEMOUNT and REJECT TAPEMOUNT commands. Labeled-tape messages are described in the *Operator Messages Manual*.

Responding to Tape Process Messages

Tape processes perform many functions, including reading the labels on mounted tapes and sending that information to $ZSVR. If a tape process encounters any problems while reading the label information on a mounted tape, it sends an error message or a warning message to the operator console.

A tape process can generate a message as a result of many conditions, including:

- A tape that is mounted on a drive has an incorrect or unexpected label.
- A tape is mounted that is incompatible with a tape process request. For example, an unlabeled tape is mounted in response to a labeled-tape request.
- A tape is mounted on a drive with manual density selection set to a density incompatible with the data on the tape.
- A tape is bad.

Tape process messages are displayed in the format:

```
$tape : { ERROR | WARNING } number - text
```

$tape

is the tape device that is reporting the error.

number

is the error number: 0001–0499 are tape process error messages, and 0901–0999 are tape process warning messages.

text

describes the error situation, containing any applicable volume serial numbers.

Example: Tape Process Warning Message

```
93-09-04 10:30:57 \SAGE.$ZSVR       TANDEM.TAPE.D20      000654 $TAPE: WARNING
                                    904 - VOLUME ABC001 DENSITY CHANGED TO 1600
                                     FROM 6250
```

What to Do if $ZSVR Abends

If $ZSVR terminates because of a fatal internal error, it automatically generates a SAVEABEND file. This file contains information about what was occurring at the time of failure and can help your management staff analyze the problem.

If the $ZSVR process terminates because of a fatal internal error:

1. Make a copy of the SAVEABEND file using the FUP DUP command.

SAVEABEND files have names of the form ZZSA*nnn* and are located on the volume and subvolume where the program file for the $ZSVR process resides. For example, if a STATUS $ZSVR showed that the program file for $ZSVR was $SYSTEM.SYS23.ZSERVER, the SAVEABEND file would be:

`$SYSTEM.SYS23.ZZSA`*`nnn`*

Then enter:

```
> FUP DUP $SYSTEM.SYS23.ZZSA001, $SYSTEM.SYS23.ZZSA002
```

2. Tell your operations manager the location of this file, or print a copy for reference.
3. Start another $ZSVR process pair:

   `> ZSERVER /NAME $ZSVR, NOWAIT, PRI 199, CPU` *`x/y`*

All pending mount messages are lost and the requester's application receives file system error 195.

Example

To copy the SAVEABEND file and begin a new $ZSVR process pair using CPUs 1 and 2, enter:

```
> FUP DUP $SYSTEM.SYS23.ZZZSA001, $SYSTEM.SYS23.ZZZSAOO2
> ZSERVER /NAME $ZSVR, NOWAIT, PRI 100, CPU 1/2
```

Responding to Tape Mount Requests

Labeled-tape messages are sent to the operator console and any other console that you have designated. All labeled-tape messages that have a sequence number require some action from you.

Note. If, for any reason, you cannot mount a requested tape, use the MEDIACOM STATUS TAPEMOUNT, DETAIL command to learn the name of the user who made the request and then notify the user of the problem.

When you receive a request to mount a labeled tape (or a scratch tape):

1. Install or remove the write ring, as requested.
2. Mount the correct tape on the requested tape drive or, if no drive is requested, on any available tape drive (use the MEDIACOM STATUS TAPEDRIVE command to determine which drives are free).
3. Ready the drive.

If the tape you mount matches the tape requested, this informational message appears:

`$ZSVR: STATUS 1504 -` *`vid`* `TAPE OPENED ON $`*`tape-device`*

If the system does not accept the tape, you get an error message.

Understanding Tape Mount Requests

When a job is ready to run in LP mode, $ZSVR sends a tape mount request to the operator console in the format:

```
$ZSVR: seq MOUNT [labeltype] vid [REEL reel]
       [DENSITY density] [ON $tape]
       {WITH RING | NO RING} [comment]
```

Requests to mount scratch tapes are displayed in the format:

```
$ZSVR: seq MOUNT [IBM] SCRATCH TAPE [REEL reel]
                 [DENSITY density]
                 [ON $tape] WITH RING [comment]
```

When you receive a message of either type, you respond to it by mounting a labeled tape according to the message elements:

seq

is the four-digit message sequence number that identifies this mount request. This number is used in the MEDIACOM REJECT TAPEMOUNT and STATUS TAPEMOUNT commands.

labeltype

asks for an IBM-standard labeled tape or a TMF tape. If *labeltype* is omitted, the request is assumed to be for an ANSI-standard labeled tape.

vid

is the volume serial number.

reel

is a number from 1 through 255 that specifies the particular reel for a tape file that extends over multiple volumes.

density

is either 1600 or 6250 bits per inch.

$*tape*

is the tape drive name in the system, displayed if you specify “device” in the tape DEFINE, or if the mount message is for a continuation volume of a multivolume operation. In this case, the tape drive used for the previous tape volume is named in the mount message and is the only drive that can be used for continuation volumes.

WITH RING | NO RING

WITH RING indicates that the tape should be write-enabled; NO RING indicates that the tape should be write-protected.

comment

is an optional message for the operator that can include information such as the relative length of the job and the degree of urgency.

Examples: Responding to Tape Mount Requests

For an ANSI-format Tape Mount Request

```
$ZSVR: 0001 MOUNT ANSABC DENSITY 1600 WITH RING
```

1. Find the ANSI-standard labeled tape with volume serial number ANSABC.
2. Install a write ring on the tape.
3. Mount the tape on any free drive that supports density 1600.
4. Ready the drive.

For an IBM-format Tape Mount Request

```
$ZSVR: 0002 MOUNT IBM IBMABC ON $TAPE NO RING
```

1. Find the IBM-standard labeled tape with volume serial number IBMABC.
2. Remove the write ring from the tape reel.
3. Mount the tape on $TAPE.
4. Ready the drive.

For an IBM-format Scratch Tape Mount Request

```
$ZSVR: 0003 MOUNT IBM SCRATCH TAPE ON $TAPE WITH RING
```

1. Find an IBM-standard labeled tape from the IBM scratch tape pool.
2. Install a write ring on the tape.
3. Mount the tape on $TAPE.
4. Ready the drive.

Responding to Tape Drive-Usage Requests

When you receive a tape drive-usage (tape mount) request, you first view the request using MEDIACOM, then respond to it by entering either the MEDIACOM ACCEPT or REJECT TAPEMOUNT command.

When you get a request to use a particular tape drive, you must accept or reject the request. The job making such a request cannot open the drive without your permission.

Viewing a Tape Drive-Usage Request

To view outstanding tape drive-usage requests, enter:

```
> MEDIACOM STATUS TAPEMOUNT [[ \node.] message-number ]
  [ , DETAIL ]
```

When you enter STATUS TAPEMOUNT without specifying a node or message number, MEDIACOM displays a brief report of all outstanding mount requests on the current default node. You can also specify the DETAIL option to receive a detailed report about all mount requests or a message you specify. See the *DSM/Tape Catalog Operator Interface (MEDIACOM) Manual* for more information.

MEDIACOM displays tape mount requests in the format:

```
Tape    Tape    Label                    Write
Mount   Name    Type   Node or Device    Prot    Action Needed
-----   ----    -----  --------------    -----   -------------

number  tname   type   nodedev           prot    action
```

number

the message number assigned by $ZSVR.

tname

the name of the tape being requested in the message.

type

the type of tape label or label processing mode (ANSI, IBM, BLP, and so on).

nodedev

the name of the node where *tname* must be mounted; or the name of the tape drive to use.

prot

specifies whether to protect *tname* from being overwritten.

action

specifies the action you must take for processing to continue, such as ACCEPT OR REJECT TAPEMOUNT.

Accepting a Tape Drive-Usage Request

To accept a tape drive-usage request, log on as a super-group user (255,*n*) and enter:

```
MC> ACCEPT TAPEMOUNT [ \node.] message-number
```

You also use the ACCEPT TAPEMOUNT command when a mounted tape cannot be read. If it is appropriate to use such a tape, the ACCEPT TAPEMOUNT command gives permission to use that tape as an unlabeled tape.

Rejecting a Tape Drive-Usage Request

To reject a tape drive-usage request, enter:

```
MC> REJECT TAPEMOUNT [ \node.] message-number
```

You can also delay the request by doing nothing.

Tape Drive-Usage Request Example

For this tape drive-usage request:

```
$ZSVR: 0004 REQUEST TO USE $TAPE UNLABELED
```

- To accept this request, log on as a super-group user (255,*n*) and enter:

  ```
  MC> ACCEPT TAPEMOUNT 4
  ```

- To reject this request, enter:

  ```
  MC> REJECT TAPEMOUNT 4
  ```

- To get more information about this request, enter:

  ```
  MC> STATUS TAPEMOUNT 4, DETAIL
  ```

A message similar to this is displayed:

```
Tape Mount          04
Open Mode           NL REQUEST
Tape Drive          $TAPE
Action              ACCEPT OR REJECT TAPEMOUNT
Tape Use            OUT
Density
Open Exclusion      EXCLUSIVE
Process Name        \SAGE.8,99
Process (CPU,PIN)   (8,99)
Define Class        NONE
Program Name        \SAGE.$SYSTEM.SYS02.BACKUP
```

Responding to Operator Attention Requests

When a tape process does not recognize a tape, the $ZSVR process sends a message to the operator console asking you to check the tape mounted on the drive. The operation cannot continue until you either accept or reject the request in the message. These requests are displayed in the format:

```
$ZSVR: seq UNKNOWN TAPE ON $tape, ACCEPT OR
           MOUNT CORRECT TAPE TO PROCEED
```

seq

is the 4-digit message sequence number that identifies this request.

$*tape*

is the tape drive on which the unrecognized tape is mounted.

When a message of this type occurs, you should either accept or reject the request as described under Responding to Tape Mount Requests on page 10-17.

If you accept an unknown tape, the system considers it to be an unlabeled tape. Use caution and make sure that the tape does not have data written at a different density.

The system might consider a tape to be unknown if:

- The tape was written at a density that the drive does not support.
- The tape has never been used.

Example

This message can be sent as the result of an unknown tape having been mounted on the tape drive named $TAPE:

```
$ZSVR:  0006 UNKNOWN TAPE ON $TAPE, ACCEPT OR
        MOUNT CORRECT TAPE TO PROCEED
```

Creating and Modifying Labeled Tapes

Use MEDIACOM to create either IBM or ANSI format labeled tapes. Using MEDIACOM, you can also display information about a tape's label, you can relabel a tape, remove a tape's label, and set whether labeled tapes are to be unloaded after being labeled or relabeled.

Labeling Tapes

You can create ANSI-standard labeled tapes using the MEDIACOM ADD TAPELABEL command. The MEDIACOM ADD TAPELABEL command, LABELS IBM option, creates IBM-MVS standard labeled tapes.

△ **Caution.** When you label a tape, all existing data on that tape becomes inaccessible. Before you label a tape, check its contents with the MEDIACOM INFO TAPELABELS command. See Displaying Tape Label Information on page 10-24.

Label a tape in ANSI or IBM format:

1. Manually write the intended label on the tape reel.
2. Log on as a super-group user (255,*n*).
3. Enter this MEDIACOM command:

   ```
   MC> ADD TAPELABEL, vid | ( vid , vid1, vid2 ) , LABELS
       IBM , TAPEDRIVE $tapedrive-name
   ```

   The system opens the tape drive in BLP (bypass label processing) mode and displays:

   ```
   $TAPE: not ready:
   ```

4. Mount the tape on the selected tape drive.

5. Ready the drive and press Return.

   If you are labeling a new blank tape, one of these messages is issued; each requires a response of either Ignore, Retry, or Abort:

```
DATA EXISTS ON TAPE
TAPE RUNAWAY ERROR FOUND, VERIFY DENSITY
DATA PARITY ERROR FOUND, VERIFY DENSITY
FILE SYSTEM ERROR ON READ: err-num
```

   Type I and press Return (or just press Return).

   This message informs you when the label operation is complete:

```
 STATUS 2501 - VOLUME vid INITIALIZED
```

   If the label operation fails, this message is displayed:

```
 STATUS 2502 - VOLUME vid NOT INITIALIZED
```

Complete syntax for the MEDIACOM ADD TAPELABEL command is included in the *DSM/Tape Catalog Operator Interface (MEDIACOM) Manual*.

Examples

- To use MEDIACOM to label an unlabeled tape with a density of 1600 bits per inch (bpi) mounted on $TAPE with the volume serial ID of ABC:

  ```
  MC> LABEL ABC, TAPEDRIVE $TAPE, DENSITY 1600
  ```

  MEDIACOM responds with messages such as:

```
$TAPE: NOT READY?
VOLUME ABC INITIALIZED
```

  The first message indicates that MEDIACOM has control of the drive and is waiting for the tape to be mounted. The second message indicates that the labeling operation is complete.

- To write the volume serial ID XYZ to an unlabeled tape mounted on $TAPE1, adding a babel which includes the owner field SOFTWARE, enter:

  ```
  MC> TAPEDRIVE $TAPE1
  MC> ADD TAPELABEL, LABELS IBM XYZ, OWNER "SOFTWARE"
  ```

- To write a standard ANSI label on an unlabeled tape and leave the tape online at the load point after labeling it, specifying a density setting of 1600 bits per inch (bpi):

  ```
  MC> ADD TAPELABEL ABC123, TAPEDRIVE $TAPE, NOUNLOAD, DENSITY
  1600
  ```

```
 $TAPE: not ready:
```

  (Mount the tape and ready the drive; press Return.)

- To write a standard ANSI label to a brand new tape, enter:

```
MC> ADD TAPELABEL ABC123, TAPEDRIVE $TAPE1, DENSITY 1600
```

```
 $TAPE1: not ready:
```

(Mount the tape and ready the drive; press Return.)

```
 VOLUME ABC123 INITIALIZED
```

- To give an unlabeled tape an IBM-MVS label with a density of 6250 bpi and the name AA0004, enter:

```
MC> ADD TAPELABEL AA0004,&
>>>    DEVICE $TD001,&
>>>    DENSITY 6250,&
>>>    LABELS IBM
```

Displaying Tape Label Information

You use the MEDIACOM INFO TAPELABELS command to determine the type of label on a tape or the contents of the first 80 bytes of an unlabeled or nonstandard labeled tape.

Note. You cannot read a 7-track tape with MEDIACOM.

To display tape label information:

1. Log on as a super-group user (255,*n*).
2. Enter this MEDIACOM command:

   ```
   MC> INFO TAPELABELS [ , TAPEDRIVE $tape ] , DETAIL
   ```

3. Mount the tape on the selected tape drive.
4. Ready the drive and press Return.
5. MEDIACOM displays one of these:
   - ANSI TAPE: Followed by two to three 80-byte records. The VOL1 record contains the volume serial number and the owner's user ID; the HDR1 label contains the creation and expiration dates.
   - IBM TAPE: Followed by the two to four 80-byte records containing the same type of information as for the ANSI format. IBM tapes can have two HDR1 records; the second one is the one to use.
   - NONSTANDARD LABELED TAPE: Followed by the first 80 bytes of the first record.

Examples

This MEDIACOM example shows how to display brief format label information for an IBM-MVS labeled tape that is mounted on the tape drive named $TAP06:

```
MC> INFO TAPELABELS, TAPEDRIVE $TAP06
```

```
Label Type                      IBM
Tape Name                       T00359
Owner                           Dept 334
Tape File
Creation Date
Expiration Date                 Scratch

1 tape label returned
```

Relabeling a Tape and Removing a Tape Label

When an application writes data to a tape, an expiration date is recorded on the tape. When this date has passed, you can relabel the tape or remove the existing label.

△ **Caution.** When you relabel or unlabel a tape, all existing data on that tape becomes inaccessible. Before you relabel or unlabel a tape, check its contents with the MEDIACOM INFO TAPELABELS command. See Displaying Tape Label Information on page 10-24.

To Relabel a Tape

1. Log on as a super-group user (255,*n*).
2. Write the intended label on the tape reel.
3. Enter this MEDIACOM command:

   ```
   MC> ADD TAPELABEL old-tape-name, NEWNAME new-tape-name,
       TAPEDRIVE $tape
   ```

   new-tape-name is the name you want to use in the relabel operation.
4. Mount the tape on the selected tape drive.
5. Ready the drive and press Return.

   When the relabeling process is complete, this message is displayed:

```
STATUS 2503 - VOLUME old-vid NEW LABEL new-vid
```

The relabeling operation changes the volume serial identification number of standard labeled tapes.

Examples: Relabeling Standard Labeled Tapes

To relabel an ANSI-format standard labeled tape, enter:

```
MC> ADD TAPELABEL, NEWNAME ABC123,CDE123,TAPEDRIVE $TAPE
```

To relabel an IBM-format standard labeled tape, enter:

```
MC> ADD TAPELABEL, LABELS IBM, NEWNAME  IBM456,IBM890,TAPEDRIVE $TAPE
```

To Remove a Tape Label

1. Log on as a super-group user (255,*n*).
2. Enter this MEDIACOM command:

   ```
   MC> DELETE TAPELABEL vid [ , TAPEDRIVE $tape ]
   ```

3. Mount a tape on the intended tape drive.
4. Ready the drive and press Return.

   A message similar to this is displayed:

STATUS 2504 - VOLUME *vid* UNLABELED

Examples: Unlabeling Tapes

To convert a standard labeled tape to an unlabeled tape, enter:

```
> MEDIACOM

MC> DELETE TAPELABEL ABC123,TAPEDRIVE $TAPE
```

To convert four uncataloged labeled scratch tapes to unlabeled tapes, enter:

```
MC> DELETE TAPELABEL (T11046, T03347, T00297, T00345),&
>>>    PROMPT OFF,&
>>>    TAPEDRIVE $TD006
```

Setting Whether Tapes Are Unloaded After Labeling

You can set whether the system unloads a tape after it performs the labeling operation using the MEDIACOM commands ENV, UNLOAD OFF, and UNLOAD ON.

When you set NOUNLOAD on, the system leaves the tape online at the load point after it completes a labeling operation. When you clear NOUNLOAD, the system unloads the tape after it completes a labeling operation.

Setting NOUNLOAD on can be helpful when you need to label a tape and then leave it mounted for a tape backup. Clearing NOUNLOAD can be useful when you are labeling many tapes.

- To display the current NOUNLOAD state by using MEDIACOM:

  ```
  MC> ENV
  ```

- To set NOUNLOAD with MEDIACOM:

  ```
  MC> UNLOAD OFF
  ```

- To clear NOUNLOAD with MEDIACOM:

  ```
  MC> UNLOAD ON
  ```

Examples

This MEDIACOM example shows that the NOUNLOAD state is on (the default value). After the UNLOAD OFF command is entered, the default value changes from on to off.

```
MC> ENV

Tape Drive          NO SESSION DEFAULT SET
File Catalog        \SKY.FILE_CATALOG_SKY
Volume Catalog      \SKY.MASTER_TAPE_CATALOG
Pool                TAPES_FOR_SKY
Volume              \SKY.$OPRVOL.OPR
Unload              ON

MC> UNLOAD OFF
MC> ENV

Tape Drive          NO SESSION DEFAULT SET
File Catalog        \SKY.FILE_CATALOG_SKY
Volume Catalog      \SKY.MASTER_TAPE_CATALOG
Pool                TAPES_FOR_SKY
Volume              \SKY.$OPRVOL.OPR
Unload              OFF
```

Tapes remain online after each labeling command until you enter another UNLOAD command or exit the session.

Guidelines

- UNLOAD ON is the default value when MEDIACOM is started.
- The UNLOAD OFF condition applies to the following commands: ADD TAPELABEL command and its options LABELS IBM, NEWNAME, and SCRATCH, and the INFO TAPELABELS command.
- The UNLOAD ON command affects only the current MEDIACOM process and is active until you change it with the UNLOAD command, OFF option.

Premounting and Scratching Labeled Tapes

System operators might be asked to premount labeled tapes in tape drives or to create scratch tapes for labeled-tape processing. This subsection describes each of these tasks and how to handle potential problems.

Premounting Labeled Tapes

All labeled tapes can be mounted before a tape mount request. When premounting a labeled tape, remember:

- Unload the premounted labeled tape if you are going to accept another request to use that drive.
- For operations that require multiple volumes, all volumes must be mounted on the same drive. Do not premount the continuation volume on another drive unless you are performing TMF backups or using the BACKUP program.
- When a premounted tape is used, you do not see the mount request for that drive.

Handling Tapes Unloaded by the System

Sometimes when you premount a labeled tape, the system unloads it. Possible solutions to some common problems include:

- If you premount a tape and it is unloaded after being placed online, look for the following error message:

```
$tape-device: ERROR 1 - CANNOT PERFORM VOLUME RECOGNITION.
                ERROR error-code
```

  - For error code 151, either the tape is blank or the density of the data on the tape is lower than the density switch setting on the drive.
  - For error code 120, the density of the data on the tape is higher than the density switch set on the drive.
  - If the tape is unloaded at all density settings, either the tape is blank or the data is written at a density that is unacceptable to the drive.

- If you mount a tape to satisfy a request and the tape is unloaded after being placed online, look for an error message in the Event Management Service (EMS) log:
  - If you used an incorrect tape, find the correct tape and mount it on the drive. If the correct tape is not available, reject the request with the MEDIACOM REJECT TAPEMOUNT command.
  - If the tape volume is not expired, and that volume was specifically requested by a user, check with the user. The wrong tape might have been requested, or you might have to scratch that tape with the MEDIACOM ADD TAPELABEL command, SCRATCH option.

- If there is no error message, use the MEDIACOM STATUS TAPEDRIVE command. If the drive is no longer “in use,” the application has been terminated and you need not recover.

- If you use a blank tape for a backup operation, the “unknown tape” message is displayed. Accept the request to use that tape drive with the MEDIACOM ACCEPT TAPEMOUNT command.

Scratching a Labeled Tape

A scratch tape is a labeled tape with an expired date. Because users often request standard labeled scratch tapes for output, you might want to keep scratch tapes apart from other tapes in your tape racks or tape library.

Note. Before you scratch an unexpired labeled tape, check with your operations management to ensure that this operation is allowed.

To create a scratch tape:

1. Log on as a super-group user (255, *n*).
2. Enter this MEDIACOM command:

   ```
   MC> ADD TAPELABEL ( tape-name [ , tape-name ] ) , SCRATCH
   ```

   The ADD TAPELABEL command is not intended for tapes cataloged in a DSM/TC system.
3. Mount the tape that has the specified *vid* on the selected tape drive.
4. Ready the drive and press Return.

Example

In this example, three tapes that contain unexpired files are changed to scratch tapes. Each tape is mounted on the default tape drive for this session.

```
MC> ADD TAPELABEL (V11233, V11244, V20055), SCRATCH
```

When the ADD TAPELABEL, SCRATCH operation successfully ends, MEDIACOM responds with a message such as:

```
TAPE VOLUME V11233 SCRATCHED
TAPE VOLUME V11244 SCRATCHED
TAPE VOLUME V20055 SCRATCHED
```

Compressing a Tape Dump File

You can compress a tape dump file using the COPYDUMP utility, which usually resides in the file $SYSTEM.SYS*nn*.COPYDUMP.

To use the COPYDUMP program to copy and compress a memory dump file from disk or tape to disk, enter:

```
> COPYDUMP source-file , dest-file
```

source-file

specifies the memory dump file that is to be copied and compressed. Specify either the name of the tape drive where the tape is loaded or the name of a disk dump file you want to copy. If *source-file* is a tape file, you must create it by performing a tape dump as described in the operator's guide for your system. If *source-file* is a disk file, you must create it by doing a bus dump as described in the operator's guide for your system or by using the RCVDUMP program or the RECEIVEDUMP command.

dest-file

specifies the destination of the COPYDUMP operation. For *dest-file*, specify the name of a disk file. If *dest-file* does not exist, it is created during the COPYDUMP operation. If *dest-file* exists, it must be empty (EOF=0) and have file code 9614.

For a complete list and description of all COPYDUMP options, see the *TACL Reference Manual*.

Note. You can also use FUP (the CREATE and COPY commands) to copy tape dump files to disk files. However, using COPYDUMP is faster, and it generates a smaller disk dump file because it compresses the dump. Also, COPYDUMP automatically determines the size of the disk dump file, whereas you must specify the extent size of the disk file if you use FUP.

Examples

Copying and Compressing a Tape Dump File

If a tape dump file resides on the tape mounted on the tape drive $TAPE2, to copy and compress the tape dump file into the disk file $DATA.DUMPS.CPU1, enter:

```
> COPYDUMP $TAPE2, $DATA.DUMPS.CPU1
```

Compressing a Disk Dump File

To compress the disk dump file $BAS10.DUMPS.CPU3 into the disk file $BAS10.CDUMPS.CPU3, enter:

```
> COPYDUMP $BAS10.DUMPS.CPU3, $BAS10.CDUMPS.CPU3
```

Solving Common Tape Subsystem Problems

For more information about common problems that can occur with tapes unloaded by the system, see Handling Tapes Unloaded by the System on page 10-28.

Table 10-3. Common Tape Subsystem Problems (page 1 of 2)

Problem	Possible Causes	Solution
Hardware:		
A device is not ready, resulting in error 100.	A tape drive was brought down or is not online.	Power up the drive and/or put it online.
A device has been brought down or a hardware failure has occurred, resulting in error 66.	A tape drive has been purposely brought down.	Bring the drive up and retry.
An interrupt timeout occurs, resulting in error 218.	An I/O process cannot communicate with a tape drive (often due to a controller failure).	Notify your management.
Software:		
File-system error 195 occurs, indicating that $ZSVR is not running.	The $ZSVR process has been purposely stopped.	Restart $ZSVR if you have the authority, or escalate the problem.
Tape:		
No write-enable ring has been installed on a tape.	An incorrect tape was mounted or a write-enable ring was not installed.	Verify that it is the correct tape, install a write-enable ring, and remount the tape.
A runaway tape was detected, or an attempt was made to access a tape whose density is lower than the switch setting on the tape drive. An error 151 results.	The system has tried to read a blank tape or a tape written at a density lower than the drive can read.	Check the tape and replace it if needed, change the density switch, or mount the tape on a different drive.
A tape continues to spin beyond the load point.	The load point has fallen off.	Use a new tape.
Every time a tape is mounted, it is unloaded.	The drive on which the tape is mounted is open for unlabeled use and a labeled tape is being mounted.	Unload the tape by using MEDIACOM or mount an unlabeled tape.
Security:		
A security violation has occurred, resulting in error 48.	An attempted operation was not allowed.	Resecure necessary files if you are authorized, or notify management.
Tape DEFINE:		
The DEFINE you specified could not be found, resulting in error 198.	The specified DEFINE name does not exist.	Correct the DEFINE name used in syntax or supply a DEFINE of the given name.

Table 10-3. Common Tape Subsystem Problems (page 2 of 2)

Problem	Possible Causes	Solution
Tape label:		
An unexpired labeled tape is being used, resulting in error 49.	An unexpired labeled tape was used.	Scratch the tape or use a different tape.
A tape label record is missing or incorrect.	An attempt was made to access a tape with an incorrect or missing label.	Use the MEDIACOM INFO TAPELABEL command, correct the label, or use a different tape.
A tape fails to respond to a BACKUP command.	A tape with an inappropriate label type was mounted in error.	Use a different tape.

11

Backing Up and Restoring Disk Information

This section describes the Backup, Restore, and Backcopy utilities:

- Backup copies disk files to tape
- Restore copies the files from the tape back to disk
- Backcopy duplicates backup tapes

For more information, including the complete syntax, guidelines, and more examples of using Backup, Restore, and Backcopy, see the *Guardian Disk and Tape Utilities Reference Manual*.

Why Use Backup and Restore?

Every organization has critical files, sets of files, or entire disks. Your system manager is usually responsible for ensuring that backups are performed on a regular basis; for example, daily, weekly, or monthly, depending on the importance of the files and the frequency with which files change.

Backup lets you copy disk files to magnetic tape, and contains many options that let you customize your backups to the needs of your site.

At many sites, a system operator backs up the disks on a regular basis. However, you might also want to use Backup and Restore to:

- Protect information stored on disk

  Back up your disk files to tape. If the original files are damaged or destroyed, you can restore them from the tape copies. Even if your system operator performs this function, you still might want to back up your important files.

- Save disk space

  Back up seldom-used disk files to tape. After the files are stored on the tape, you can delete them from disk. If the files are needed, you can restore them from the tape.

- Move files from one system to another system

  For systems that are not connected on a network, you can back up files on one system and then restore them on another system.

- Prevent accidental overwriting

  Labeled tapes can provide protection from accidental overwriting of information.

- Convert files from DP1 format to DP2 format, or vice versa

  Backup and Restore have options to perform disk-process file conversion. You can back up files in one format and then restore them in another format.

Supported Modes of Operation

Backup and Restore support the following modes of operation:

- File mode. Backup and Restore copy individual files one at a time. A backup tape consists of a sequence of individual files. You can back up any file for which you have read access.

- Volume mode. Backup and Restore copy an entire disk volume. Only a super ID (255,255) can use this mode. Backcopy cannot duplicate backup tapes created in volume mode.

This guide discusses file mode operations. For information about volume mode operations, see the *Guardian Disk and Tape Utilities Reference Manual.*

Backing Up Your Files

To back up all or selected files on your disk using file-mode operation:

1. Prepare a backup tape. This tape must be write-enabled. For a tape reel, insert a write-ring; for a cartridge tape, turn the SAFE arrow to point away from the word SAFE.
2. Locate an available tape drive, but do not mount the tape until you start the Backup process.
3. Enter a Backup command at the TACL prompt:

```
BACKUP [ / run-options / ] tape-drive,
qualified-file-set-list
[ , command-option ]
[ , command-option ]...
```

run-options

One or more run options. See the RUN command in the *TACL Reference Manual.*

tape-drive

Name of a local or remote tape drive, or a TAPE DEFINE name.

qualified-file-set-list

One or more file-set lists (including any qualifiers) specifying the files you want to back up.

command-option

Backup command option. See Table 11-2.

For example, this Backup command contains only the required elements (the keyword Backup, the tape drive name, and a qualified file-set list), and copies all files from the $DATA.USER subvolume to the tape on the tape drive $TAPE1:

```
10> BACKUP $TAPE1, $DATA.USER.*
```

Backup attempts to access the tape drive $TAPE1. If the drive is available, Backup takes control of the drive, displays its program banner, and prompts you for a tape:

```
BACKUP Program - T9074D46 (07SEP98)
Copyright Tandem Computers Incorporated 1981-1998
$TAPE1: Not Ready?
```

The program banner includes the name of the program, the product number and release version, the release date, and the copyright notice.

The “Not ready?” message indicates that Backup has control of tape drive $TAPE1 and is waiting for you to mount your tape on this drive. To cancel the Backup operation, enter CTRL-Y at the “Not ready?” prompt; you will exit the Backup program and return to the TACL program.

If the tape drive is in use by another process, Backup returns file-system error 12 (file in use) and stops. If this happens, select another tape drive and start again.

4. Mount your write-enabled tape according to the directions for the specific tape drive. Press Return.

   Backup copies the files from $DATA.USER to the tape on $TAPE1.

This procedure ensures that you have control of the tape drive before you put your tape online. If you mount your tape before you enter your Backup command, the Backup operation begins immediately without the "Not ready?" message. If the drive is already controlled by another process, this process could write to your tape.

Specifying a File-Set List for Backup

You can back up any files on your system to which you have read access. To back up:

- A single file, specify the file name in your Backup command.
- More than one file, specify a file set or file-set list.

A file set is one or more related files in the format:

```
[\system.][$volume.][subvolume.]file-id
```

If you omit the *system*, $*volume*, or *subvolume*, Backup uses the current default system, volume, or subvolume. For example, if you supply only a volume name followed by a file identifier, Backup fills in the subvolume name.

Note. Avoid using subvolume defaulting. This feature might not be supported in future releases of Backup.

You can include these wild-card characters in the *volume*, *subvolume*, and *filename*:

* (asterisk)	Matches zero to eight character in the same position where it appears
? (question mark)	Matches any single character in the position where it appears

These characters cannot be used in a system name or for the volume identifier ($) or field separator (.). Also, some Backup command options do not allow these characters; see the *Guardian Disk and Tape Utilities Reference Manual*.

Examples of file sets are:

$MYVOL.EXTRA.INFO	The file INFO in the subvolume EXTRA on volume $MYVOL.
MYFILE	The file MYFILE on the current default subvolume and volume.
.	All files on the current default subvolume.
$Z*.SUB*.*	All files in subvolumes that have names beginning with SUB on every disk volume that has a name beginning with $Z.

*.CASH	Files named CASH in every subvolume on the current default volume.
MAIL.MESS*	Files that have names beginning with MESS on the MAIL subvolume of the current default volume.
VOL*.?FILE	Files that have names five characters long ending with FILE, that reside on any subvolume that has a name beginning with VOL.

This command backs up all the files in the current default volume and subvolume:

```
10> BACKUP $TAPE, *
```

Because no volume or subvolume name is given in this example, Backup uses the current default values for volume and subvolume.

To back up all the files in the volume $MANUF:

```
11> BACKUP $TAPE, $MANUF.*.*
```

To back up all files to which you have read access in the current default system:

```
12> BACKUP $TAPE, *.*.*
```

This command includes a file set containing special-case wild cards. This Backup command copies all files in the current default system that meet these requirements: the file name can be any length but must begin with SD, and the subvolume name must be five letters long ending with MON:

```
14> BACKUP $TAPE, ??MON.SD*
```

A file-set list is one or more file sets. To include more than one file set in a file-set list, separate the file sets with commas and enclose them in parentheses. This example shows a file-set list in a command that backs up all files in the current default subvolume and all the files in the volume $MANUF:

```
15> BACKUP $TAPE, (*, $MANUF.*.*)
```

Using a Qualified File-Set List

You can use a qualified file-set list to specify criteria for including or excluding files or SQL objects in a file set. A qualified file-set list is a file-set list that includes one of the qualifiers from the following table. (A simple file-set list does not include a qualifier.) You can specify a qualifier only once for each file-set list.

For a detailed description of file-set qualifiers, see the *Guardian Disk and Tape Utilities Reference Manual*.

Table 11-1. File-Set List Qualifiers

Qualifier	Function
EXCLUDE	Excludes files from the file-set list
FROM CATALOG[S]	Specifies SQL objects in an SQL catalog
START	Specifies the starting file in a file-set list
WHERE	Includes or excludes files from a file-set list

This example backs up all files on $DISK2 that are owned by the user with user ID 8,76:

```
10> BACKUP $TAPE1,$DISK2.*.* WHERE OWNER = 8,76,LISTALL
```

This example backs up all EDIT files (file code 101) on $DISKA.SALESVOL and $DISKB.REPORTS that are owned by users in the group SALES:

```
11> BACKUP $TAPE1,($DISKA.SALESVOL.*,$DISKB.REPORTS.*)&
11> &WHERE OWNER = SALES.* AND FILECODE = 101,LISTALL
```

Using Run Options in a Backup Command

A Backup command can also include any run option that is valid for the TACL RUN command. Run options control the operation of the process you are starting. For a complete list of run options, see the RUN command in the *TACL Reference Manual*.

The OUT file option sends the Backup listing to a disk file or a spooler location, such as a printer. You can specify an existing disk file or a nonexistent file for the OUT file. If you specify a nonexistent file, Backup creates an entry-sequenced file with that name.

For example, to send the Backup listing to the file LIST2 in the current default subvolume:

```
10> BACKUP / OUT LIST2 / $TAPE2, $DATA.USER.*
```

The IN file option specifies a file or device from which Backup reads the command parameters. If your Backup command contains a long or complicated list of files and command parameters, entering the command from an IN file can be easier than entering it interactively.

Instead of typing the parameters when you enter your Backup command, type the parameters in a command file. Then, when you enter the Backup command, specify this file for the IN file parameter.

Note. You must use an IN file if your Backup command exceeds 132 characters for a single command line or 528 characters for a command that continues beyond one command line.

For example, this command starts a Backup process that uses the parameters in the IN command file FRED.FLIST and sends its listing to the OUT file BRECD. With the

NOWAIT option, control of your terminal returns to your TACL process during the back up operation:

```
11> BACKUP / IN FRED.FLIST, OUT BRECD, NOWAIT /
```

The file FLIST contains Backup command parameters such as:

```
$TAPE2,
( BOOKS.*,
MOVIES.*,
.
.
.
MAGS.*) ,
LISTALL
```

Using Backup Command Options

The Backup command accepts one or more command options. Some of the common options, such as DENSITY, NOT, LISTALL, and PARTIAL, are described here. All options are described in the *Guardian Disk and Tape Utilities Reference Manual.*

Table 11-2. Backup Command Options (page 1 of 2)

Option	Function
ALTFILE	Changes the name of an alternate-key file in the file label of a primary-key file
ARCHIVEFORMAT	Specifies archive tape format (tape version 3)
AUDITED	Backs up files that are audited by the Transaction Monitoring Facility (TMF)
BLOCKSIZE	Sets the size of tape records (blocks) written to tape
DENSITY	Sets the recording density
DP1FORMAT	Creates tape files in DP1 format
DP2FORMAT	Creates tape files in DP2 format
DSLACK	Specifies minimum slack space for data blocks
EXT	Specifies extent sizes for the destination file in a file conversion
IGNORE	Ignores most data errors encountered
INDEXES	Specifies whether indexes defined on SQL tables are backed up with the tables
ISLACK	Specifies minimum slack space for index blocks
LISTALL	Lists all files specified in the file-set list, including any files that cause errors
MSGONLOCK	Requests, when Backup encounters an open file, that Backup prompt the user to indicate whether to skip the file or wait for it to become available

Table 11-2. Backup Command Options (page 2 of 2)

Option	Function
MULTIDRIVE	Lets up to four tape drives be queued for unlabeled-tape backup operations
NOMYID	Disallows the Restore MYID option
NOPROMPT	Instructs Backup not to prompt the user before beginning to write on each tape, but to begin when it detects the tape drive is ready
NOSAFEGUARD	Excludes Safeguard information from the back up tape
NOSQLDATA	Makes Backup skip data transfer for all SQL files in the file set
NOT	Specifies files that are not to be backed up
NOUNLOAD	Means that the final tape is rewound and left online (rather than offline) when the back up finishes
OPEN	Backs up open files (however, if a file is open for write access, the backup of the file might not be usable when restored)
PAGELENGTH	Lets users specify between 20 and 100 lines per page to be displayed on a utility listing (default value is 60)
PART	Renames a secondary partition in a partitioned file
PARTIAL	Backs up only files modified since a specified date
PARTONLY	If ON, lets parts of a file be backed up without backing up all of the files' partitions, based on selection criteria; if OFF, backs up an entire partitioned file
REMOTEIOSIZE	Specifies the maximum size of each data block transfered between systems
SCRATCHVOL	Selects an alternate disk volume for the temporary files that Backup creates during a file conversion
SHAREOPEN	Causes backup files to be opened with shared-read access instead of read/write access
SQLCATALOGS	Lets you back up SQL catalog tables
START	Selects a starting point in the file-set list for back up
TAPEMODE	Specifies the tape recording mode for streaming tape units
VERIFYREEL	Validates each file on the tape after the reel is completed; This includes checking tape-record sequence numbers, and reading the volume labels, file labels, data records, and checksums
VOL	Gives new volume and subvolume names for files on tape
VOLUMEMODE	Performs volume-mode back up (restricted to super ID 255,255)

Specifying Backup Recording Density

If you are using a tape drive that supports the density selection, you can specify one of three recording densities (in bits per inch):

- 6250 bpi
- 1600 bpi
- 800 bpi (G-series only)

To specify a recording density, include the DENSITY option and a density setting in your Backup command. For example:

```
10> BACKUP $TAPE1, $RIVER.RUN.*, DENSITY 1600
```

The tape drive then generates a tape with the recording density you specified. If you do not specify a density, the tape receives the current density setting of the tape drive.

Listing All Files During Backup

Backup lists or displays the names of the files that are backed up. By default, the Backup listing includes only files that generate error or warning messages. Backup, however, can list all files that are backed up, in addition to any error messages, if you use the LISTALL option.

The Backup listing is displayed at the OUT file you name in your Backup command; if you omit the OUT option, Backup uses your terminal. Regardless of the listing device you specify, messages (including errors and instructions) are also sent to the terminal where you entered the Backup command.

This example shows a listing generated by Backup without the LISTALL option:

```
3> BACKUP $TAPE3, *
```

Backup displays:

```
BACKUP Program - T9074D46 (07SEP98)
Copyright Tandem Computers Incorporated 1981-1998
Tape: $TAPE3  Operating System: G06  Tape Version: 1
Backup options: BLOCKSIZE 8, NO PARTONLY
Backup time: 11Feb1999  10:43                                    Page: 1

Reel: 1      Code        EOF    Last modif   Owner RWEP   Type   Rec Block

$MYVOL.MYSUBVOL
 BFILE        *ERROR*  File aborted (Open error 12).
 XFILE        *ERROR*  File aborted (Open error 48).

Summary Information
Files dumped =  9  Files not dumped = 2
```

In the above example, two attempts to copy files resulted in errors. The error number in parentheses identifies a file-system error. For example, error 48 is a security violation; the user did not have read access to that file.

Include the LISTALL option to get a complete listing of the files backed up and the files that generate errors or warnings. This example shows the output from a Backup command with the LISTALL option:

```
4> BACKUP $TAPE3, *, LISTALL
```

Backup displays:

```
BACKUP Program - T9074D46 (07SEP98)
Copyright Tandem Computers Incorporated 1981-1998
Drives: $TAPE3
System: \LAUN11  Operating System: G06  Tape Version: 1
Backup options: BLOCKSIZE 8, NO PARTONLY
Backup time: 11Feb1999 10:43                                        Page: 1

Tape: 1        Code      EOF    Last modif     Owner  RWEP  Type Rec  Bl

$MYVOL.MYSUBVOL
 BFILE        *ERROR*  File aborted (Open error 12).
 CFILE          101    334525  6Jan1999  8:55   8,44  NUNU
 CFILE2         101      1854  8Jan1999  9:19   8,44  AOGO
 EFILE                   5120  8Jan1999 10:46   1,25  NUNU   E   80  1024
 FILE1          101     22657 15Jan1999 15:11   1,25  CUCU

 Mount next tape #2?

Tape: 2        Code      EOF    Last modif     Owner  RWEP Type Rec   Bl
 FILE2          101      1118 15Jan1999 11:12   1,25  CUCU
 FILE3          101    365520 27Jan1999 14:37   1,25  CUCU
 INDEX          101     61616 19Jan1999 19:05   8,12  CUUU
 KEYFILE                32768  3Dec1998 10:08   1,25  AOAO   K   80  4096
 SEC1           101     26374 18Jan1999  9:05   1,25  CUUU

 XFILE        *ERROR*  File aborted (Open error 48).

Summary Information
Files dumped =  9 Files not dumped = 2
```

In this example, the Backup operation requires two tapes. Backup displays the prompt “Mount Next Tapes #2?”when the first tape is full. After you mount another tape, press Return to continue the Backup process.

With the LISTALL option, you can create a permanent record of the files that were copied. In your Backup command, include the LISTALL option and name a disk file or a spooler location for the OUT option. You can label the contents of the tape by attaching the output to the reel or cartridge.

Omitting Files From a Backup

To omit some files from the backup, specify them with the NOT option. For example, this command backs up all files in subvolume $CAFFE.MED except MITSLAG and MOKA:

```
5> VOLUME $CAFFE.MED
6> BACKUP $TAPE1, * , NOT (MITSLAG, MOKA)
```

Backing Up Only Changed Files

Use the PARTIAL option to back up only files that have been modified since a given date and time. Enter PARTIAL followed by the date (that is, the first three letters of the month, the day of the month, and the year, with spaces separating them), a comma, and the time (using a 24-hour clock).

For example, this command backs up all files in the system that were modified after 6 p.m. on January 24, 1999:

```
7> BACKUP $TAPE1, *.*.*, PARTIAL JAN 24 1999, 18:00
```

If you regularly back up your files, you can use the PARTIAL option to save tape space and backup time. Each time you back up files, use the PARTIAL option to specify the date and time of the previous backup. This lets you maintain current backup copies of all your files without copying files that are already backed up.

Restoring Your Files

Use Restore to copy backed up files from a backup tape to a disk. Restore contains many options that let you customize your restore operations, including:

- Copy files to disk from a magnetic tape created using Backup
- List the contents of a tape without restoring data
- Convert files from one disk process type to another

Note. Do not run Restore under a user ID with Safeguard default protection, because all files restored to disk would also have Safeguard protection.

For more information about tape operations, see Section 10, Using Labeled Tapes.

This example restores all files on tape drive $TAPE1 that have the current default volume and subvolume name and all files that have the volume and subvolume name $DATA.USER.

To restore files from a backup tape to your disk using file-mode operation:

1. Remove the write-ring from the tape reel that you are using for the restore operation. This write-protects your tape; if your tape is write-enabled and another process attempts to write to your tape, your data can be damaged or destroyed.
2. Enter a Restore command at the TACL prompt:

```
RESTORE [ / run-options / ] tape-drive,
qualified-file-set-list
[ , command-option ]
[ , command-option ]...
```

run-options

One or more run options. See the RUN command in the *TACL Reference Manual.*

tape-drive

Name of the tape drive you are using for the restore operation, or a TAPE DEFINE name. Tape drive names begin with a dollar sign ($) and can be up to seven characters long.

qualified-file-set-list

One or more file-set lists specifying the files you want to restore.

command-option

Restore command option. See Table 11-3 on page 11-14.

This example shows a Restore command:

```
10> RESTORE /OUT LIST, NOWAIT/ $TAPE1, & (*, $DATA.USER.*), LISTALL
```

Restore prompts you for tape drive $TAPE1 with the “Not Ready?”message.

3. Mount your tape and press Return. Restore copies the specified files from tape to disk.

The above command includes the OUT option to send the listing to the file LIST and the NOWAIT option so you can use your terminal for other applications while Restore is running. The Restore command LISTALL option causes Restore to list all files that are restored (including any files that cause an error).

To restore a file from tape that has the same name as a file on disk, you must have purge access to the file on disk, unless you specify the KEEP option.

Using Run Options in a Restore Command

A Restore command can include any run option that is valid for the TACL RUN command. Run options control the operation of the process you are starting. For a complete list of run options, see the RUN command in the *TACL Reference Manual*.

The OUT file option sends the Restore listing to a disk file or a spooler location, such as a printer. You can specify an existing disk file or a nonexistent file for the OUT file. If you specify a nonexistent file, Restore creates an entry-sequenced file with that name.

For example, this command sends the Restore listing to the file LIST2 in the current default subvolume:

```
10> RESTORE / OUT LIST2 / $TAPE2, $DATA.USER.*
```

The IN file option specifies a file or device from which Backup reads the command parameters. If your Restore command contains a long or complicated list of files and command parameters, entering the command from an IN file can be easier than entering it interactively.

Instead of typing the parameters when you enter your Restore command, type the parameters in a command file. Then, when you enter the Restore command, specify this file for the IN file parameter.

Note. You must use an IN file if your Restore command exceeds 132 characters for a single command line or 528 characters for a command that continues beyond one command line.

Using Restore Command Options

Restore accepts one or more command options. Some of the common options are described here. All options are described in the *Guardian Disk and Tape Utilities Reference Manual*.

Table 11-3. Restore Command Options (page 1 of 2)

Option	Function
AUTOCREATECATALOG	Specifies whether nonexistent SQL catalogs are created automatically
ALTFILE	Changes the name of an alternate-key file in the file label of a primary-key file
AUDITED	Restores audited files
CATALOG[S]	Specifies the catalog(s) that describe SQL objects
DETAIL	Provides details about each file in the file-set list
DSLACK	Sets the minimum slack space for data blocks in file conversions
ISLACK	Sets the minimum slack space for index blocks in file conversions
INDEXES	Specifies whether indexes defined on SQL tables are automatically restored
EXT	Sets extent sizes for the destination file in a file conversion
IGNORE	Ignores most data errors encountered
KEEP	Does not restore the file if it already resides on the disk
LISTALL	Lists all files specified in the file-set list, including any files that cause errors
LISTONLY	Lists files on the tape without actually restoring any files
MAP NAMES	Changes destination disk file names
MULTIDRIVE	Allows up to four tape drives to be queued for unlabeled-tape backup operations
MYID	Gives restored files to the user ID running Restore
NOPROMPT	Instructs Restore not to prompt the user before beginning to read each tape, but to begin when it detects the drive is ready
NOSAFEGUARD	Excludes Safeguard security information
NOT	Excludes the specified file or files from the restore
NOUNLOAD	Rewinds the tape and leaves it online after restore
OPEN	Restores backed up files that were open at the time of the backup (however, if a file was open for write access, the restored file might not be usable)
PAGELENGTH	Lets users specify between 20 and 100 lines per page to be displayed on a utility listing (default value is 60)
PART	Renames secondary partitions of partitioned files
PARTOF	Restores only those partitions whose primary partitions reside on a specific volume

Table 11-3. Restore Command Options (page 2 of 2)

Option	Function
PARTONLY	Restores only partitions that are specified in the file-set list
REBUILD	In a set of tapes, restores the latest version of each file that was on the first back up tape
REMOTEIOSIZE	Specifies the maximum size of each data block transfered between systems
SCRATCHVOL	Selects an alternate disk volume for temporary files that Restore creates during a file conversion (DP1 or DP2)
SQLCATALOGS	Specifies whether SQL catalogs are to be restored
SQLCOMPILE	Specifies whether an SQL object program is automatically recompiled during a restore
SQLTAPEPARTARRAY	Makes Restore update all target object partitions with the partition array from the tape
START	Selects the point in a file-set list where the restore is to begin
TAPEDATE	Gives a restored file the same modicification date as the file on the tape
TURNOFFAUDIT	Restores files that were audited by TMF and restores audited files as nonaudited files
VERIFYTAPE	Verifies the volume label, file label, and data records without actually restoring any files (used with LISTONLY)
VERIFY	Verifies the restored data
VOL	Specifies the destination volume for restored files

Listing All Restored Files

Restore lists files in the same format Backup does, and both normally list only those files for which errors or warnings are generated. You can, however, get a listing of all the files restored to disk by including the LISTALL option in your Restore command.

For example, this command restores all the files in the subvolume MAIZE on disk volume $CORN and sends a list of all files restored to the OUT file $CORN.MEAL:

```
10> RESTORE / OUT $CORN.MEAL / $TAPE, & $CORN.MAIZE.*, LISTALL
```

Listing Files Without Restoring Them

To get a list of files on a single backup tape without actually restoring the files, use the LISTONLY option:

```
10> RESTORE /OUT LIST/ $TAPE1, *.*.*, LISTONLY, NOUNLOAD
```

Restore sends the listing to the file LIST. The NOUNLOAD option keeps your tape online and ready to begin a restore operation after the listing is finished.

After Restore finishes, LIST contains this listing:

```
RESTORE Program - T9074D46 (07SEP98)
Copyright Tandem Computers Incorporated 1981-1998
Tape: $TAPE1  Operating System: G06  Tape Version: 1
Backup options: BLOCKSIZE 8, NO PARTONLY
Restore (list only) time: 12Feb1999 12:07  Backup time: 11Feb1999 10:43
Page: 1

Reel: 1        Code         EOF    Last modif   Owner  RWEP Type  Rec Bl

$MYVOL.MYSUBVOL
 CFILE          101      334525  6Jan1999  8:55  8,44  NUNU
 CFILE2         101        1854  8Jan1999  9:19  8,44  AOGO
 EFILE                     5120  8Jan1999 10:46  1,25  NUNU  E    80  1024
 FILE1          101       22657 15Jan1999 15:11  1,25  CUCU
 FILE2          101        1118 10Jan1999 11:12  1,25  CUCU
 FILE3          101      365520 27Jan1999 14:37  1,25  CUCU
 INDEX          101       61616 19Jan1999 19:05  8,12  CUUU
 KEYFILE                  32768  3Dec1998 10:08  1,25  AOAO  K    80  4096
 SEC1           101       26374 18Jan1999  9:05  1,25  CUUU

Summary Information
Files on tape =  9
```

Preserving Current Disk Files During Restore

The KEEP option preserves files that are currently on disk when you restore files that have the same name. If you do not include the KEEP option, a disk file is replaced by the file on the tape. For an example, see Specifying Files to Not Restore on page 11-16.

Assigning Your ID to All Restored Files

The MYID option specifies that the user ID of the user who runs the Restore program becomes the owner of all the files restored. Also, the security of all restored files is set to that person's log-on default security. If the MYID option is omitted, each restored file belongs to the user who owned it when it was backed up, and the file security is the same as when the file was backed up. A backup tape generated using the NOMYID option cannot be restored using the MYID option.

For an example, see Specifying Files to Not Restore on page 11-16.

Specifying Files to Not Restore

Use the NOT option in a Restore command to specify files on the backup tape that you do not want to restore. The example below does the following:

- Restores all the files on tape that were backed up from volume $PASTRY except the files in subvolumes LINZER and PETIT4
- Keeps any files currently on disk that have the same name as tape files specified in the Restore command file-set list
- Gives you ownership of the restored files

The Restore command is:

```
10> VOLUME $PASTRY
11> RESTORE $TAPE1, *.*, NOT (LINZER.*, PETIT4.*), KEEP, MYID
```

Restore displays:

```
RESTORE Program - T9074D46 (07SEP98)
Copyright Tandem Computers Incorporated 1981-1998
Tape: $TAPE1  Operating System: G06  Tape Version: 1
Backup options: BLOCKSIZE 8, NO PARTONLY
Restore time: 12Feb1999 16:13  Backup time: 10Feb1999 13:38        Page: 1

Reel: 1        Code      EOF    Last modif    Owner  RWEP   Type   Rec Bl

$PASTRY.PUFF
 APPLE          101    61616 19Jan1999 19:05   8,44  CUCU
 CHEESE         101     1854  8Jan1999  9:19   8,44  CUCU
 CHERRY         101   334525  6Jan1999  8:55   8,44  CUCU

$PASTRY.DOUGH
 FILO           101    22657 15Jan1999 15:11   8,44  CUCU
 COOKIE         101     1118 20Jan1999 11:12   8,44  CUCU
 BREAD          101   365520 27Jan1999 14:37   8,44  CUCU

Summary Information
Files restored = 6   Files not restored = 0
```

Specifying New Restored File Destinations

By default, Restore restores files to the volume and subvolume whose names match the volume and subvolume names of the files on tape. To restore files to a different volume or subvolume, use the MAP NAMES option. Any parts of the file name that you do not specify remain the same as the tape file names.

For example, if you want to restore the file-set:

```
($TOLSTOY.NOVELS.*, $PROUST.*.*)
```

To restore the files to the subvolume $BIG.BOOKS:

```
10> RESTORE $TAPE, ($TOLSTOY.NOVELS.*,$PROUST.*.*),&
10> &MAP NAMES ($TOLSTOY.NOVELS.* TO $BIG.BOOKS.*, &
10> &$PROUST.*.* TO $BIG.BOOKS.*)
```

You can also use the MAP NAMES option with Restore to convert files from one disk-process format to another. For example, you can convert files from DP1 format to DP2 format (that is, restore DP1 files to a DP2 disk volume), by specifying a DP2 destination volume with the MAP NAMES option.

If you use the MAP NAMES or VOL option and your backup tape has two (or more) files with the same file name, only the last file with the same name is restored. For example, consider this file-set list:

```
($TOLSTOY.NOVELS.BIG1, $PROUST.RTP.BIG1)
```

Suppose you attempt to restore both of these files to the subvolume $GIANT.NOVELS by entering:

```
20> RESTORE $TAPE, ($TOLSTOY.NOVELS.BIG1, $PROUST.RTP.BIG1), VOL $GIANT.NOVELS
```

Because files are stored on tape in alphabetical order, the data in $PROUST.RTP.BIG1 is first restored to $GIANT.NOVELS.BIG1 but then is purged and overwritten with the data in $TOLSTOY.NOVELS.BIG1. After the restore finishes, $GIANT.NOVELS.BIG1 contains only the data from the $TOLSTOY.NOVELS.BIG1 tape file.

Using Labeled Tapes With Backup and Restore

If labeled-tape processing is enabled for your system, you can use Backup and Restore with both labeled and unlabeled tapes.

To access a file on a labeled tape, you must use a TAPE DEFINE — a named set of attributes and values that specifies information about a tape file such as the volume ID, tape density, and operator mount messages.

Backup can use all the listed attributes in a TAPE DEFINE for labeled tapes. Restore, however, can use only DEVICE, MOUNTMSG, SYSTEM, and VOLUME. If your DEFINE contains other attributes, Restore ignores them.

Table 11-4. TAPE DEFINE Attributes for Backup and Restore

Attribute	Function
DENSITY	Tape density in bits per inch (bpi)
DEVICE	Name of the tape drive on which all tapes must be mounted
EXPIRATION	Expiration date for the tape file
FILEID	Tape file name (17-character string)
GEN	Integer value (in the range 0001-9999) that indicates the generation number
LABELS	Label processing mode; BACKUP, IBMBACKUP, or OMITTED for Backup; or BACKUP, IBMBACKUP, OMITTED, or BYPASS for Restore
OWNER	Owner ID (1-14 characters) in the tape label
MOUNTMSG	Mount message to be displayed with the system mount message or tape drive use request
RETENTION	Retention period in number of days for the tape file (the default is 1)
SYSTEM	Name of the system that controls the tape drive on which the tape (or tapes) must be mounted
TAPEMODE	Mode (STARTSTOP or STREAM) for a cartridge tape drive
USE	Use of the tape file (input or output)
VERSION	Integer value (in the range 00-99) that indicates the version number within one generation
VOLUME	Six-byte tape volume ID for the backup tape, or SCRATCH to indicate that any scratch tape is acceptable for the backup tape

For more information about DEFINEs and labeled tapes, see Section 6, Creating and Using DEFINEs, and Section 10, Using Labeled Tapes.

Using a TAPE DEFINE With Backup

This example shows how to back up the files in disk volume $DISK2.WEEK to two labeled backup tapes with volume IDs TP022 and TP023.

1. Create the TAPE DEFINE (here named =MYBACKUP) to be used for your backup, using the ADD DEFINE command to describe the labeled tape.
2. Confirm the TAPE DEFINE attributes and their values by using the INFO DEFINE command to display information about =MYBACKUP.

```
10> ADD DEFINE =MYBACKUP, CLASS TAPE, LABELS BACKUP,&
10> &VOLUME (TP022, TP023)
11> INFO DEFINE =MYBACKUP
Define Name         =MYBACKUP
CLASS               TAPE
VOLUME              (TP022, TP023)
```

3. Enter a Backup command using the TAPE DEFINE named =MYBACKUP.

```
3> BACKUP =MYBACKUP, $DISK2.WEEK.* , LISTALL
```

   Backup displays a mount request for this job at the operator console:

```
$ZSVR: 0005 MOUNT TP022 WITH RING
"Labeled BACKUP. Tape #1."
```

4. You or the operator mount the requested tape.

   Because the DEFINE named =MYBACKUP does not include a DEVICE attribute, the two tapes for this Backup operation can be mounted, in order, on any available drive.

   The system checks the DEFINE attributes with the mounted tape. The Backup operation starts after the correct tape is mounted (in this case, a labeled backup tape with volume ID TP022).

Backup sends this listing to the OUT file, which in this case is the home terminal of the Backup process.

```
BACKUP Program - T9074D46 (07SEP98)
Copyright Tandem Computers Incorporated 1981-1998
*WARNING*  This tape can only be restored with TNS/II RESTORE (C00 or
           later).
Labeled BACKUP tape. Operating System: G06  Tape Version: 3

Backup options: NO AUDITED, BLOCKSIZE 8, NO IGNORE, NO OPEN, NO PARTONLY

Backup time: 12Feb1999 22:34                                      Page: 1

Tape: 1      Code       EOF    Last modif  Owner   RWEP  Type  Rec Bl

*First tape*  Tape #1, volume id: TP022, drive: $TAPE.

$DISK.WEEK
 BUILD        101     46282  5Feb1998 21:11  1,3   CUCU
 COMMENTS     101      5512 10Feb1998 18:08  1,3   CUCU
 COPY         101    128858 27Jan1998  2:38  1,3   CUCU
 COVER        101      2540 28Jan1998 18:09  1,3   CUCU
 CPU1       *Continued on next tape*
 CPU1       *Tape change*  Tape #2, volume id: TPO23, drive: $TAPE
 CPU1                        5Feb1998  9:33  1,3   CUCU
 DIST         101      5364 10Feb1998 17:05  1,3   CUCU
 INTERN1      101     75234 12Dec1998 20:21  1,3   CUCU
 TRFTEST               7502 10Feb1998 17:25  1,3   CUCU

Tape volumes used: TP022, TP023

Summary Information
Files dumped = 8  Files not dumped = 0
```

Using a TAPE DEFINE With Restore

To restore tapes created with Backup using a TAPE DEFINE, do one of the following:

- Use the original TAPE DEFINE in your Restore command.
- Modify the original DEFINE, using the ALTER DEFINE command, and use that DEFINE in your Restore command.
- Create a new DEFINE and use it in your Restore command.
- Do not use a TAPE DEFINE. Instead, follow the standard procedure for running Restore described under Restoring Your Files on page 11-12.

Restore can access the files on any backup tape using any of these options. To choose which option to use, determine which is most convenient for you to use for each situation in which you must restore files.

This example illustrates the second option, showing how to modify the DEFINE you created for your backup command. In this example, you add a mount message and a device name to the TAPE DEFINE =MYBACKUP used in the previous Backup example.

1. View the attributes of the existing Backup DEFINE, using the INFO DEFINE command, as illustrated below.
2. Add a mount message and a device name to the TAPE DEFINE, using the ALTER DEFINE command, as illustrated below.
3. Enter a Restore command using the TAPE DEFINE you have altered.

```
10> INFO DEFINE =MYBACKUP , DETAIL
Define Name           =MYBACKUP
CLASS                 TAPE
VOLUME                (TP022, TP023)
LABELS                BACKUP
11> ALTER DEFINE =MYBACKUP, DEVICE $TAPE0, MOUNTMSG &
11> &"Emergency recovery of $DISK2.WEEK"
12> RESTORE =MYBACKUP,$DISK2.WEEK.*,LISTALL,TAPEDATE
```

Restore displays this mount message at the operator console:

```
$ZSVR: 0006 MOUNT TP022 ON $TAPE0 NO RING
"Labeled RESTORE. Tape #1. Emergency recovery of $DISK2.WEEK"
```

4. Mount the first tape in the set (with volume ID TP022) on tape drive $TAPE0.

   The Restore operation begins. When the restoration of TP022 is finished, Restore prompts for the second tape, if necessary. The second tape must also be mounted on $TAPE0.

Duplicating Backup Tapes With Backcopy

Backcopy lets you make one or two duplicate copies of tapes produced by the Backup program in file mode. Backcopy does not duplicate tapes produced by Backup in volume mode, or system image tapes (SITs).

There are several reasons to duplicate your backup tapes:

- Disaster recovery for disk files

  If you duplicate a backup tape, you can store each copy in a different location. Then, if a disaster, such as a fire or flood, occurs at one location, the duplicate tape is safe at the other location.

- Distribution of disk files

  After copying files to tape with the Backup program, you can make duplicate copies of this tape for distribution to other systems. This procedure is especially useful for systems that are not connected on a network.

- Archival of disk files

  After duplicating a backup tape, you can keep a copy of its content archived in case you ever need to use its content again. This is true not just for disaster recovery, as described above, but also in the event that you later need information from files you at one point thought you would no longer need on your system.

Running Backcopy

When you run Backcopy, you can make one or two duplicate tapes. If you make one duplicate copy, Backcopy generates a tape in the same format as the original tape unless:

- You use labeled tapes.
- You specify the ARCHIVEFORMAT option.

For these cases, Backcopy generates a tape in the archive tape format.

If you make two duplicate copies, Backcopy always generates the tape in tape format 3 (archive tape format).

Entering a Backcopy Command

You enter a Backcopy command at the TACL prompt or from a command file:

```
BACKCOPY [ / run-options / ] source-tape, dest-tape, *.*.*
[ , command-option ]
[ , command-option ]...
```

run-options

One or more run options. See the RUN command in the *TACL Reference Manual.*

source-tape

Name of the tape drive that reads the tape you are duplicating, or a TAPE DEFINE name.

dest-tape

Name of the tape drive(s) where the duplicate tape(s) is written, or a TAPE DEFINE name.

..*

Specifies all files on the source tape. You cannot specify individual files.

command-option

A Backcopy command option:

Table 11-5. Backcopy Command Options

Option	Function
ARCHIVEFORMAT	Specifies the archive tape format (tape version 3)
DENSITY	Specifies the tape density in bits per inch (bpi)
LISTALL	Lists all files copied, including any files that cause errors
NOUNLOADIN	Causes the last source tape not to be unloaded
NOUNLOADOUT	Causes the last destination tape (or tapes) not to be unloaded
PAGELENGTH	Lets users specify between 20 and 100 lines per page to be displayed on a utility listing (default value is 60)
TAPEMODE	Specifies the tape mode for streaming tape drives
VERIFYREEL	Makes Backcopy verify each tape reel after the reel is written

Backcopy copies all files. If Backcopy cannot read a file because of a parity or checksum error, it displays an error message and the tape duplication fails.

A duplicate tape generated with Backcopy is different from a parallel tape copy generated with Backup:

- When parallel copies are made with Backup, both copies contain exactly the same amount of information on each reel. You mount new tape reels for both copies at the same time. Backup makes identical copies so that each reel of tape is interchangeable with its corresponding copy during a Restore.
- When you make a second copy with Backcopy, the amount of information on each tape reel can vary between the copies. You mount a new tape whenever a drive reaches the end of reel; you do not necessarily change tape reels for both copies at the same time. The individual reels are not interchangeable; only the whole duplicate tape set is interchangeable with the original set.

Backcopy Examples

Making one copy of an unlabeled tape

This Backcopy command makes one copy of an unlabeled backup tape. The source tape is on tape drive $TAPE1, and the destination tape is on tape drive $TAPE2. The format of the new tape is the same format as the source tape. The LISTALL option causes Backcopy to list all files that are copied.

```
10> BACKCOPY $TAPE1, $TAPE2, *.*.*, LISTALL
```

Making two copies of an unlabeled tape

This Backcopy command makes two copies of an unlabeled backup tape. The source tape is on tape drive $TAPE1, and the destination tapes are on tape drives $TAPE2 and $TAPE3. The format of the new tapes is the archive tape format.

```
11> BACKCOPY $TAPE1, ($TAPE2, $TAPE3), *.*.*
```

Making one copy of a labeled tape

This Backcopy command makes one copy of a labeled tape with volume identification HD004. The ADD DEFINE command sets the attributes for this labeled tape. The source tape is on tape drive $TAPE1; Backcopy displays a message for you to mount the destination tape. The destination tape is unlabeled because the TAPE DEFINE does not define it. The format of the new tape is the archive tape format.

```
12> ADD DEFINE =Source ^Tape, CLASS TAPE, &
12> &LABELS BACKUP, VOLUME HD004
13> BACKCOPY =Source ^Tape, $TAPE2, *.*.*
```

Making one copy of an unlabeled tape

This Backcopy command makes one copy of an unlabeled tape on $TAPE1. The destination tape is labeled because the TAPE DEFINE defines it.

```
12> ADD DEFINE =Output, CLASS TAPE, LABELS BACKUP, &
12> &VOLUME SCRATCH
13> BACKCOPY $TAPE1, =Output, *.*.*
```

Part IV. Using the Spooler and Its Utilities

This part of the guide contains information about the spooler subsystem, and describes using Peruse and Spoolcom to manage spooler jobs and the spooler, respectively:

- Section 12, Introduction to the Spooler
- Section 13, Managing Your Spooler Jobs Using Peruse
- Section 14, Performing Routine Spooler Operations Using Spoolcom
- Section 15, Managing the Spooler Using Spoolcom

12 Introduction to the Spooler

The Compaq *NonStop*™ Kernel spooler is a set of programs that acts as an interface between your application programs and the printers on your system.

When you run a NonStop™ Kernel program and print something:

1. You send the output to the spooler.
2. The spooler receives the output, stores it as a spooler job in a print queue on disk, and monitors the status of the printer.
3. The spooler keeps the spooler job in the print queue until the designated printer becomes available.
4. The spooler sends the output to the printer to be printed.

The NonStop™ Kernel spooler:

- Keeps operating even if a processor or disk fails.
- Lets you change the destination of a job after it enters the spooler. For example, you can send a job to a holding location on the spooler print queue and then select a printer while the job is on the queue.
- Lets you monitor and control the status of your jobs, examine a spooler job before you print or delete it, and change the printer for your output. No programming is necessary; send your output to the spooler simply by specifying a spooler location as your OUT file.
- Lets an operator monitor and control all spooler components.

You access the spooler by running either:

- Spoolcom (the spooler management utility), the primary tool that system operators and managers use for interactive control of the spooler
- Peruse, which lets all users view and control their own print jobs

Topic	Page

Why Use the Spooler?

Different applications (in addition to your own) can send output to the same printer at the same time. The spooler saves each job on the print queue and sends them to the printer depending on their priority.

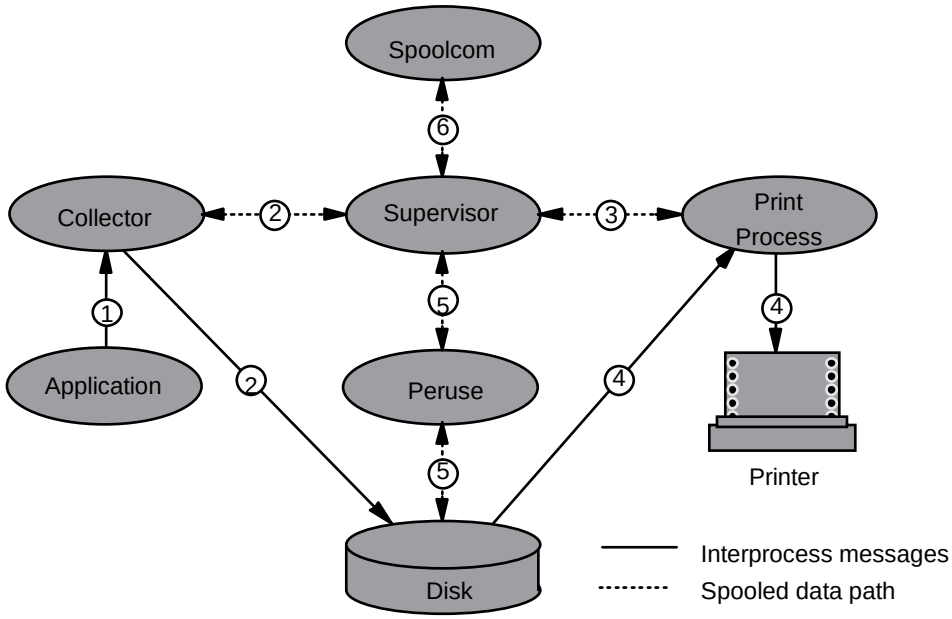
The spooler protects your applications from any device-dependent considerations. You don’t need to know the technical specifications for each printer. For example, to print a job on a specific printer, you only need to know the spooler location associated with the printer (such as $S.#LP for a line printer).

Spooler Components

The spooler consists of processes, output devices, and routing structures:

Spooler Term	Definition
Spooler supervisor	Process (usually named $SPLS) that monitors and communicates with the other spooler processes and controls printing of spooler jobs. Each spooler has one supervisor.
Collector process	Process that accepts output from applications and stores it on disk. Each spooler must include at least one, but no more than 15, collectors.
Print process	Process that retrieves a spooler job stored on disk by a collector process and prints the job to an output device. Each printer in the spooler has one print process associated with it; however, each print process can control several output devices.
Device	Physical device (printer), process, or virtual device to which output is directed. Output devices are components of the system, not of the spooler.
Routing structure	Structure that directs jobs to output devices and consists of a set of locations and print devices.
Job	Set of data sent to the spooler to be printed.
Peruse	Utility that lets you control and monitor your jobs. You enter Peruse commands interactively from your terminal. See Section 13, Managing Your Spooler Jobs Using Peruse.
Spoolcom	Utility that lets a system operator create, initialize, and get status for spooler components, set job attributes, and start a printer that is offline. See Section 14, Performing Routine Spooler Operations Using Spoolcom.

Figure 12-1. How Spooler Components Interact



1. The **application** sends its output to the collector process.
2. The **collector** writes data to the disk and informs the supervisor that it has accepted a job.
3. The **supervisor** informs the print process where the job can be found on the disk.
4. The **print process** reads the job from disk and writes the job to the printer.
5. **Peruse** obtains the job information from the supervisor or lets you examine data on the disk.
6. **Spoolcom** permits system operators to modify the status of spooler components.

CDT 008.CDD

Spooler Jobs and Job Attributes

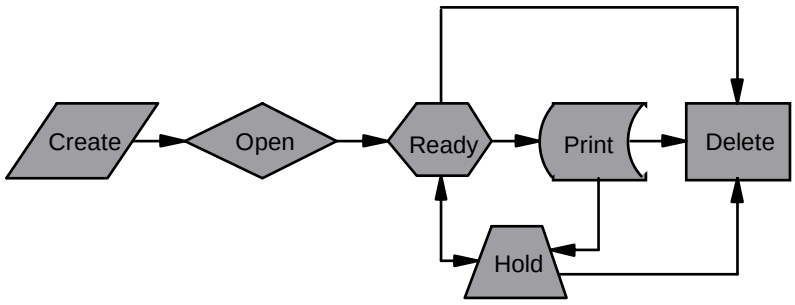
When you send output from an application to the spooler, the output is called a spooler job. Spooler job attributes are assigned characteristics for a particular job. To display these attributes for your spooler jobs, use the Peruse STATUS or JOB command (see Section 13, Managing Your Spooler Jobs Using Peruse, for details).

A spooler job file is an unstructured disk file with file code 129 that contains a spooler job, including print data records and formatting control information. You use Peruse commands to manage your spooler job files like you manage your spooler jobs.

Attribute	Definition
Batch Number	Identifies a batch, which is a group of individual jobs that have been linked together with the Peruse LINK command.
Job Number	A unique number in the range 1 through 4095 that the spooler assigns to identify each spooler job.
Job Copies	Specifies the number of copies that the spooler should print.
Job Form Name	An optional attribute that lets you guarantee that your job is printed only on the printer that has the same form name. For example, if an application produces a job that fills out W-2 forms, the job must be printed on a printer loaded with W-2 form paper. If the job has a form name of “W2,” it will print only on a printer that has the same form name. The form name “W2” is assigned to the printer when the special paper is loaded.
Job Priority	Determines when a job will print in relation to other jobs queued for the same printer. The spooler maintains a queue for all printers. Higher-priority jobs are placed near the front of the queue, while lower-priority jobs are placed near the end of the queue. See Selection Algorithm on page 12-6.
Job State	Describes the job status (see Figure 12-2 on page 12-5), which can be: **OPEN**: Job being added to the spooler. For example, if you send a TFORM process’ output to the spooler, the job is OPEN until the TFORM output is complete. **READY**: Job ready to print, but not yet printing (usually because another job is printing). **PRINT**: Job being printed. PRINT is normally the last state before the job is deleted from the spooler. To prevent the spooler from deleting a job after printing, use Peruse to set the hold-after-printing flag; see Section 13, Managing Your Spooler Jobs Using Peruse, for details. **HOLD**: Job not printed, remains in the spooler indefinitely until you delete it or remove the hold. You can put a job on HOLD using the Peruse HOLD command. You can put a job on hold at any time. If the job is in READY or PRINT, it is placed on hold immediately. If it is in OPEN, it is placed on hold after the job is completely spooled.

Attribute	Definition
Job Report Name	The job report name is printed in the header message of the job, which is described under Printer Attributes on page 12-6.
Job Location	The job location can be associated with a physical printer on the system or with a dummy holding location. See Routing Structure on page 12-8.
Life Cycle of a Job	1. A job starts at the OPEN state while the application is writing the data to the collector process. 2. The collector process stores the data as a job in a print queue on disk. 3. When the application is finished, the job is in the READY state. 4. The job enters the PRINT state if its spooler location is associated with a printer. 5. The print process controls the printing of the job. 6. When printing is complete, the spooler deletes the job, unless the hold-after-printing flag is set.

Figure 12-2. Life Cycle of a Spooler Job



Create: You create a job when your application opens a collector process and writes to it; for example:

TFORM /IN *filename,* OUT $S.*device/*

Open: The application writes to the collector process, which in turn writes the data to a disk file.

Ready: The job is ready when the application closes the collector process.

Hold: You can hold a job, removing it from the device queue, only when it is in the ready or print state.

Print: The job waits in the device queue until it is ready to print. While in the print state, the job is printing on the output device. If you place the job on hold, it immediately stops printing.

Delete: If the hold-after-printing flag is not set, the job is deleted after it finishes printing. You can also delete a job from the spooler using a specific request. If the job is printing when you make that request, the job immediately stops printing and is deleted.

CDT 009.CDD

Printer Attributes

Each printer (and associated print process) on your system has these attributes that affect your spooler jobs:

Form Name

The form name of a printer defines the type of job that can be printed on it. Only a job whose form name is the same as the form name of the printer can be printed. For an example, see Job Form Name on page 12-4.

Header Message

The header message includes the job report name, location, job number, form name, and date and time of printing. An operator can turn on or off the header message for each printer. When the header message is on, it prints on the first page of the job; the report name and location are printed in large letters (see Figure 12-3). If the header message is off, jobs print consecutively with only a form feed (new page) to indicate the beginning of the next job.

If the system operator specifies a batch header, the job information prints on two of the three trailer pages as well as on the first two pages of each job. The trailer pages have printing over the page folds, enabling jobs printed on accordion-fold paper to be easily separated. The two-page header message always appears on the top page, regardless of how the job is folded.

The actual header message depends on the print process controlling the printer. The headers described above are produced by a NonStop™ Kernel print process. If a printer is controlled by a user-written print process, it can produce a custom header (or none at all).

State

The printer state describes the status of the printer:

Printing	The printer is printing a job.
Waiting	The printer is idle and waiting for a job to print.
Offline	The printer is offline and not available for printing.
Suspended	The printer is in the process of printing a job but has been suspended by the operator (for example, to change ribbons).
Deverror	The printer received a file-system error while printing. Operator intervention is required.
Procerror	The supervisor process determined that the print process for the printer is not working correctly. Operator intervention is required.

Selection Algorithm

For each printer, the spooler maintains a queue, which is a list of the jobs to be printed on the printer. The job at the head of the queue is the next job to be printed.

Jobs with a high priority usually print sooner than lower priority jobs. However, a selection algorithm affects the order in which jobs print within the same priority level.

If the selection algorithm is FIFO (First In, First Out), jobs are placed at the end of the queue and wait to be printed. If the selection algorithm is not FIFO, the spooler lets short jobs print before longer jobs of the same priority.

For a description of the spooler job-selection algorithm, see Controlling Jobs on page 14-20.

Figure 12-3. Sample Header Page

Routing Structure

The spooler routing structure, which consists of a set of locations and printers, directs a spooler job to a printer.

A spooler location is the logical destination of a job, while a printer is the physical destination. This distinction allows flexibility when routing jobs. The spooler assigns each job a location when it enters the spooler. The job eventually prints on the printer associated with that location, if a printer exists with that name.

Location names have two parts: a group name and a destination name. The group name is always preceded by a number (#) symbol. Examples of location names are:

```
#LP.EAST
#LP.WEST
```

#LP is a group name; EAST and WEST are destination names.

Broadcast and Nonbroadcast Groups

If you specify only the group name #LP as the location, the spooler supplies the destination. If the group is a nonbroadcast group, the spooler routes the job to the destination that can print the job soonest. If the group is a broadcast group, the job is routed to all of the destinations in the group and prints on all the printers associated with the group.

For example, assume that the location name #LP.EAST is associated with a line printer on the east side of the machine room, and #LP.WEST is associated with a line printer on the west side of the machine room.

If #LP is a broadcast group, a job routed to #LP prints at both line printers, and two copies of the job are printed. If #LP is a nonbroadcast group, the first available line printer prints the job, and only one copy of the job is printed. In either case, a job routed to #LP.WEST will print once at the line printer at the west end of the machine room.

Your system operator can tell you the locations available to you, the printer associated with each location, and the groups that are broadcast groups.

Default Routing

The spooler has a special location, #DEFAULT, which is used when you do not specify a location for a job. For example, these two commands are equivalent; both send TFORM output to $S.#DEFAULT:

```
10> TFORM / IN DAYREPRT, OUT $S /
11> TFORM / IN DAYREPRT, OUT $S.#DEFAULT /
```

Ask your system manager which physical printer or printers are associated with #DEFAULT on your system.

Implicit Route Creation

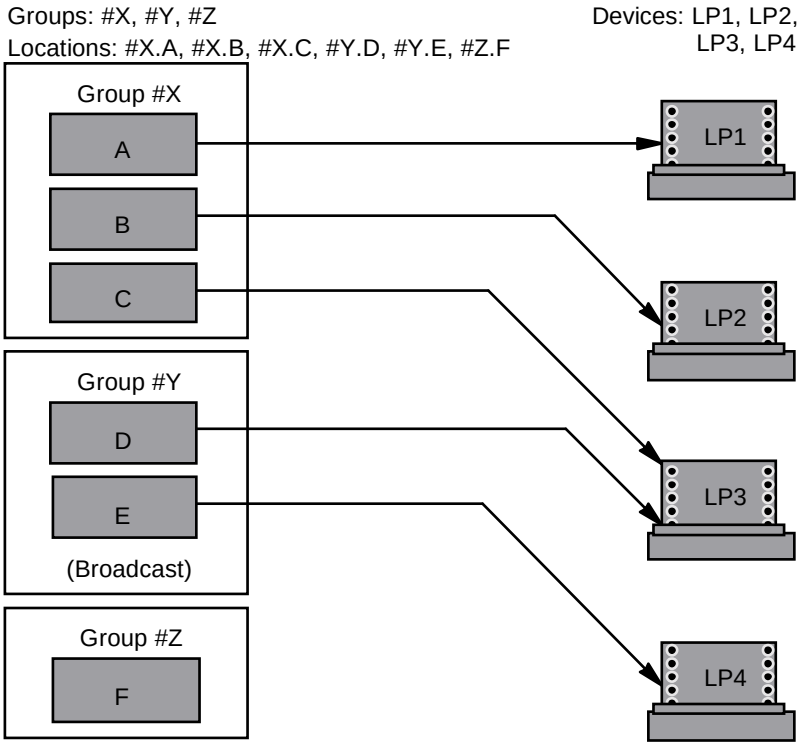
When jobs are routed to nonexistent locations or groups, the spooler implicitly creates a route:

- When a job is routed to #LOC.1, where either the group #LOC doesn't exist or the location #LOC.1 doesn't exist, the spooler creates the location #LOC.1.
- A job routed to a nonexistent group, #LOC, results in the creation of location #LOC.DEFAULT.

You can use the location #HOLD (or any other nonexistent location) as a holding location so that you can use Peruse to examine your job before it is printed.

Figure 12-4 shows the association of spooler locations with actual printers.

Figure 12-4. Spooler Routing Structure



A printer can be connected to several locations, but each location is connected to no more than one printer.

Jobs routed to #X print on one of LP1, LP2, or LP3.
Jobs routed to #X.B print on LP2.
Jobs routed to #Y print on both LP3 and LP4.
Jobs routed to #Z stay in the spooling system indefinitely. They can be printed by changing the location.

Connections are established or changed by the system operator.

CDT 011CDD

Printing To the Spooler

To use the spooler from the TACL program, you specify a spooler destination for the OUT file parameter when you run your application. The spooler destination can either be a spooler location or a SPOOL DEFINE.

Sending Output to a Spooler Location

You can send your output directly to the spooler by designating a collector (such as $S) and a location (such as #LP.WEST) as the OUT file when you run your application. For example, to send the output from a TFORM process to the spooler:

```
10> TFORM /IN MYFILE, OUT $S.#LP.WEST/
```

The collector $S creates the spooler job, assigns a job number, and stores the output in a disk file. When TFORM is finished, the spooler puts the job in the READY state.

If a printer is associated with #LP.WEST, the job is printed when the printer is available. If a printer is not associated with the location, the job remains on the print queue until you delete it or send it to another printer.

The command is still valid if you do not specify the entire location name:

```
11> TFORM /IN MYFILE, OUT $S.#LP/
```

If #LP is a nonbroadcast group, the job is printed on the first available printer associated with the #LP group.

For location groups that have only one associated printer, the full location name is unnecessary. This is a valid TFORM command that does not specify the location:

```
12> TFORM /IN MYFILE, OUT $S/
```

In this case, the job is sent to #DEFAULT and is printed on one of the printers associated with the #DEFAULT group name.

Sending Output to a SPOOL DEFINE

You can use a SPOOL DEFINE to specify the attributes of a spooler job before you create the job. A DEFINE is a named set of attributes and values that you can use to specify information for a process before you start the process. You can use a SPOOL DEFINE wherever a spooler collector process name can be used.

A SPOOL DEFINE allows you to set parameters for a spooler job before you run the process that creates the job. A SPOOL DEFINE associates spooler job attributes, such as copies, report name, and location, with a SPOOL DEFINE name such as =SPOOLER^JOB. When you run the process that uses the SPOOL DEFINE, the output from that process has the job attributes you specified in the SPOOL DEFINE.

For an introduction to DEFINEs, see Section 6, Creating and Using DEFINEs.

Table 12-1. SPOOL DEFINE Attributes

Attribute	Function
BATCHNAME	Batch name for a job run under NetBatch control
COPIES	Number of copies to be printed
FORM	Form name indicating a requirement (such as special paper) for a job
HOLD	Sets hold-before-printing flag for a job
HOLDAFTER	Sets hold-after-printing flag for a job
LOC	Spooler location
MAXPRINTLINES	Maximum number of lines per job
MAXPRINTPAGES	Maximum number of pages per job
OWNER	Owner of a job
REPORT	Report name to be printed in the job header
SELPRI	Selection priority of a job

This example shows how to use a SPOOL DEFINE to set the attributes of a spooler job. First, ensure that DEFINEs are enabled for your TACL process (that is, the DEFMODE setting is ON):

```
10> SHOW DEFMODE
        Defmode  ON
```

If the DEFMODE is set to OFF, enter a SET DEFMODE ON command.

Creating a SPOOL DEFINE

1. Enter an ADD DEFINE command to create a SPOOL DEFINE named =PAY-RUN and to set the attributes:

```
11> ADD DEFINE =PAY-RUN, CLASS SPOOL, COPIES 3, &
11> &FORM PAYLST, LOC $S.#LLP, &
11> &OWNER PAYROLL.MANAGER, SELPRI 7,&
11> &REPORT "January Payroll"
```

2. Enter an INFO DEFINE command with the DETAIL parameter to display the attributes and their values for =PAY-RUN.

```
12> INFO DEFINE =PAY-RUN, DETAIL
Define Name           =PAY-RUN
CLASS                 SPOOL
LOC                   $S.#LLP
COPIES                3
FORM                  PAYLST
OWNER                 8,255
SELPRI                7
REPORT                JANUARY PAYROLL
```

Using a SPOOL DEFINE

1. Run the program that produces the spooler output.

   In this example, the program PAYLIST prints a payroll listing from the January payroll records. The OUT run option is set to the SPOOL DEFINE you just created.

```
13> RUN PAYLIST / IN PAYRCDS, OUT =PAY-RUN, NOWAIT /
```

   The spooler job created by PAYLIST has the attributes specified in the SPOOL DEFINE named =PAY-RUN.

2. To see these spooler job attributes, use Peruse:

```
14> PERUSE
PERUSE - T9101D10 - (08JUN92)    SYSTEM  \WEST
Copyright Tandem Computers Incorporated 1978, 1982, 1983, 1984, 1985, 1986,
1987, 1988, 1989, 1991, 1992

  JOB  BATCH STATE PAGES COPIES PRI HOLD LOCATION  REPORT
  380        READY 63    3      7        #LLP      JANUARY PAYROLL
_
```

Altering a SPOOL DEFINE

1. Suppose you want to run another payroll job with the PAYLIST program, but you want different values for the LOC, COPIES, and HOLDAFTER attributes. Enter an ALTER DEFINE command to change these values:

```
20> ALTER DEFINE =PAY-RUN, LOC $S.#LAZR, COPIES 1, HOLDAFTER ON
```

2. Enter an INFO DEFINE command to display the new values.

```
21> INFO DEFINE =PAY-RUN, DETAIL
Define Name           =PAY-RUN
CLASS                 SPOOL
LOC                   $S.#LAZR
COPIES                1
FORM                  PAYCHK
HOLDAFTER             ON
OWNER                 8,255
SELPRI                7
REPORT                JANUARY PAYROLL
```

3. Run the PAYLIST program; the spooler job now has these new values.

Deleting a SPOOL DEFINE

1. If you no longer need the =PAY-RUN DEFINE (and don’t want to wait for the TACL program to delete it when you log off), you can delete it as shown below:

```
22> DELETE DEFINE =PAY-RUN
```

2. Enter an INFO DEFINE command for =PAY-RUN.

   The TACL program displays this message:

```
23> INFO DEFINE =PAY-RUN, DETAIL
    Define Name        =PAY-RUN
DEFINE does not exist""
```

13

Managing Your Spooler Jobs Using Peruse

Peruse is the program you use to manage your print jobs in the spooler subsystem.

Using Peruse, you can:

- Examine a spooler job before printing or deleting it
- Monitor a job while it is being spooled
- Alter the job attributes, such as the location associated with a printer, number of copies, or report name
- Print or display specific pages or all of a spooled job
- Copy a spooler job from the spooler queue to a spooler job file (file code 129) or an EDIT file (file code 101)
- Copy a spooler job file to the spooler queue

For more information about Peruse, see the *Spooler Utilities Reference Manual*.

Running Peruse

To run Peruse, enter the keyword PERUSE at your TACL prompt. Peruse displays its program banner, copyright message, and prompt, which is an underscore (_):

```
10> PERUSE
PERUSE - T9101D10 - (08JUN92)    SYSTEM  \WEST
Copyright Tandem Computers Incorporated 1978, 1982, 1983, 1984, 1985, 1986,
1987, 1988, 1989, 1990, 1991
_
```

Fields in the Peruse banner are:

- T9101D10 is the product number and version of Peruse.
- (08JUN92) is the release date for this version of Peruse.
- \WEST is the system where this Peruse process is running.

If you start Peruse on another system, Peruse also displays the SPOOLER SUPERVISOR IS message that contains the system and spooler supervisor names.

Spooler Jobs

If you have jobs on the spooler queue when you start Peruse, these jobs are listed below the program banner as shown below:

```
PERUSE - T9101D10 - (08JUN92)    SYSTEM  \WEST
Copyright Tandem Computers Incorporated 1978, 1982, 1983, 1984, 1985, 1986,
1987, 1988, 1989, 1990, 1991

  JOB  BATCH STATE PAGES COPIES PRI HOLD LOCATION     REPORT
  456        PRINT 16    1      4        #LPP         ACCOUNTS
  555        OPEN        1      4    B   #DEFAULT     REPORTS
  1435       READY 30    1      4    A   #HOLD        JANUARY
```

Each line under the column headers describes a different job. The column headers are:

JOB — Job number of each job, as assigned by the spooler supervisor process. If a spooler job file is the current job, Peruse displays FILE under this header.

BATCH — Spooler batch job number to which this spooler job is linked. For single jobs, no batch number is listed.

STATE — Status of each job:

- OPEN — The job is still being collected by the spooler.
- READY — The spooler has finished collecting; the job is queued and waiting to print.
- HOLD — The hold-before-printing flag is on.
- PRINT — The job is currently printing.

PAGES — Number of pages in each job. OPEN jobs are being collected, so the number of pages is not known.

COPIES Number of copies of the job to be printed.

PRI Priority of the job. The range is 0-7, with 0 being the lowest; the default is 4.

HOLD Hold status of the job:

- A The hold-after-printing flag is on. The job remains on the queue after it is printed.
- B The hold-before-printing flag is on, but the job cannot be placed in HOLD (the job is OPEN).
- X An error occurred for the job (for example, a process abended while spooling the job).
- blank There is no hold on the job.

LOCATION Spooler location of the job. If the location is associated with a printer and the job is ready, the job is queued for printing. Otherwise, the job remains on the queue, and you can use Peruse to examine, print, or delete it.

If a spooler job file is the current job, Peruse displays the file name under LOCATION and REPORT.

REPORT Report name that is printed in the job header message.

You can generate the Peruse display described above at any time during a Peruse session by entering the JOB command at the Peruse prompt.

Entering Peruse Commands

After displaying your jobs, Peruse displays its prompt, an underscore (_). You can enter one Peruse command per command line, or you can enter two or more commands on the same line, if you separate each command with a semicolon (;). The maximum length of a Peruse command line is 132 characters, and each line must end with a Return.

You can usually abbreviate a Peruse command, such as J for JOB and E for EXIT. These examples show abbreviated Peruse commands:

```
_J 123
_DEL
_E
```

The above commands are the same as:

```
_J 123; DEL; E
```

If you know the job number, you can also enter these commands at the TACL prompt. A semicolon (;) must precede each Peruse command, as shown above. The spaces before and after the semicolons are optional.

```
10> PERUSE ; J 123 ; DEL
```

Declaring the Current Job

Most Peruse commands affect only the current job. After starting Peruse, you declare the current job using one of the following methods:

- Set the current job explicitly by entering a JOB command with the job number. (You can also set the job by entering only the job number.)
- Set the current job implicitly by entering a Peruse command, or by pressing Return or a function key. The most recently spooled job becomes the current job. For example, if you enter a LIST command, the most recently spooled job becomes the current job, and Peruse lists data from that job.

  You cannot set the current job implicitly with the DEL command. To delete a job, first set the job as the current job using one of the above methods, and then issue the DEL command.

After you set the current job, Peruse identifies this job in subsequent displays with a J to the left of the job number. In the following example, job number 1435 is the current job:

```
PERUSE - T9101D10 - (08JUN92)    SYSTEM  \WEST
Copyright Tandem Computers Incorporated 1978, 1982, 1983, 1984, 1985, 1986,
1987, 1988, 1989, 1990, 1991

  JOB  BATCH STATE PAGES COPIES PRI HOLD LOCATION      REPORT
  555        OPEN        1      4    B   #DEFAULT      REPORTS
J 1435       READY 30    1      4    A   #HOLD         JANUARY
```

You declare a spooler job file as the current job using the JOB command. For example, if you have a spooler job file named JOBFILE in your current subvolume, enter a JOB command and the file name (and a second JOB command to invoke the Peruse display):

```
PERUSE - T9101D10 - (08JUN92)    SYSTEM  \WEST
Copyright Tandem Computers Incorporated 1978, 1982, 1983, 1984, 1985, 1986,
1987, 1988, 1989, 1990, 1991

  JOB  BATCH STATE PAGES COPIES PRI HOLD LOCATION      REPORT
  555        OPEN        1      4    B   #DEFAULT      REPORTS
J 1435       READY 30    1      4    A   #HOLD         JANUARY
_JOB JOBFILE
_JOB

  JOB  BATCH STATE PAGES COPIES PRI HOLD LOCATION      REPORT
  FILE             1     1      4        $DISK2.SUBVOL.JOBFILE
  555        OPEN        1      4    B   #DEFAULT      REPORTS
J 1435       READY 30    1      4    A   #HOLD         JANUARY
```

Displaying a Job

There are three ways to display the current spooler job using Peruse:

- Use the LIST command.
- Press Return to display one line. Press and hold down Return to scroll the job until you release the key.
- Use function keys to list groups of lines from the job. The number of lines listed by each function key is shown below:

Key	Lines	Key	Lines	Key	Lines	Key	Lines
F1	2	F5	32	F9	512	F13	8132
F2	4	F6	64	F10	1024	F14	16384
F3	8	F7	128	F11	2048	F15	32768
F4	16	F8	256	F12	4096	F16	65536

Using the Break Key

You can use the Break key in Peruse as follows:

- If Peruse is waiting for a command, pressing Break wakes the TACL program and displays the TACL prompt. If you enter the PAUSE command at the TACL prompt, Peruse regains control of your terminal and displays its prompt. If you enter the STOP command at the TACL prompt, the TACL program stops the Peruse process (provided Peruse was the last process you started).
- If Peruse is listing lines from a job, pressing Break stops the listing and redisplays the Peruse prompt (_).
- If the STATUS command is executing, pressing Break stops its execution and redisplays the Peruse prompt (_).

Peruse Commands

For a detailed description of these Peruse commands, see the *Spooler Utilities Reference Manual*.

Table 13-1. Peruse Commands

Command	Function
BATCH	Sets a current batch job number
COPIES	Alters the number of copies for the current job
DEL	Deletes the current job
DEV	Displays the status of a printer
EXIT	Terminates the Peruse session
FC	Allows you to recall, change, and reexecute a Peruse command
FILES	Displays all file names in the current subvolume or a specific subvolume; spooler job files are preceded by a period (.)
FIND	Finds an occurrence of a string in the current job
FORM	Changes the form name of the current job
HELP	Displays the syntax for Peruse commands
HOLD	Sets the hold-before-printing flag for the current job
HOLDAFTER	Sets the hold-after-printing flag for the current job
JOB	Displays information about a spooler job or job file; sets the current job
LINK	Links a spooler job with a spooler batch job
LIST	Copies pages from the current job to a process, terminal, printer, or disk file
LOC	Changes the routing location of the current job
NUMCOL	Sets the number of columns displayed by LIST
OPEN	Specifies a new spooler supervisor
OWNER	Changes the owner of the current job
PAGE	Changes and displays the page position of the current job
PRI	Changes the printing priority of the current job
PURGE	Deletes a file in either your default subvolume or a specified subvolume
REPORT	Changes the report name of the current job
SJFILES	Displays all spooler job file names in the current or a specific subvolume
STARTCOL	Sets the first column to be displayed by LIST
STATUS	Monitors and displays the status of spooled jobs
UNLINK	Separates a spooler job from a spooler batch job
VOLUME	Sets the default volume and subvolume

Using Peruse With TFORM

This subsection describes how to use Peruse with a spooler job generated by TFORM.

Generating Your Spooler Job

Run TFORM and send the spooler job to a temporary holding location so you can examine the job before it is printed. You can choose any name for a temporary holding location if the name is not associated with a printer and is a legal spooler location name:

```
10> TFORM /IN MEMO, OUT $S.#HOLD, NOWAIT/
```

Start Peruse and examine your jobs on the queue:

```
12> PERUSE
PERUSE - T9101D10 - (08JUN92)    SYSTEM  \WEST
Copyright Tandem Computers Incorporated 1978, 1982, 1983, 1984, 1985, 1986,
1987, 1988, 1989, 1990, 1991

  JOB  BATCH STATE PAGES COPIES PRI HOLD LOCATION      REPORT
  534        READY 20    1      4        #HOLD         USERS JOHN
```

Finding a Key Phrase in Your Spooler Job

Use the FIND and PAGE commands to locate specific pages in a spooler job. For example, if your spooler job contains only one occurrence of the string ANNUAL DUES, first use the FIND command to locate the string, and then use the PAGE command to display the page number:

```
_FIND /ANNUAL DUES/
        ANNUAL DUES.              $  3488.00
_PAGE
PAGE: 14     LINE: 12
```

Use the LIST command to print page 14.

Finding TFORM Errors

You can also use the FIND and LIST commands to locate TFORM errors in a spooler job. Because all TFORM errors begin with the string *** ERRORS, use the FIND command to locate this string, and then use the LIST command to display the error:

```
_FIND /*** ERRORS/
        *** ERRORS, PAGE 1 ***
_LIST *
\outlength 80
 *
LINE 1     - COMMAND ERROR
```

When you enter the LIST command, use an asterisk (*) for the page number. Peruse displays the page from the previous FIND command, which contains the error.

Continue using the FIND and LIST commands to find any remaining errors; if you don't specify a new string, FIND continues to search for the same string (*** ERRORS). After you have found all errors, the FIND command displays only the Peruse prompt.

After making a note of your errors, you can delete the job with the DEL command, exit Peruse, and correct your TFORM document.

Altering Job Attributes

You can use several Peruse commands to change the attributes of your spooler jobs. You can change the number of copies to be printed, change the location for a job, or change the report name with the COPIES, LOC, and REPORT commands.

For example, this command string increases the number of copies to 10, changes the location to #HT09, and sets the report name to MEMO:

```
_COPIES 10; LOC #HT09; REPORT MEMO
```

The JOB command shows that the changes have been made to the job:

```
_JOB

  JOB  BATCH STATE PAGES COPIES PRI HOLD LOCATION     REPORT
J 534        READY 20    10     4        #HT09        MEMO
```

Printing Part of a Job

You can print selected pages of a spooler job with the LIST command. For example, the following command prints the first eight pages and page 14 of job 534:

```
_LIST /OUT $S.#LP3/ 1/8, 14 C
_JOB

JOB  BATCH STATE PAGES COPIES PRI HOLD LOCATION     REPORT
J 534        READY 20    10     4        #HT09        MEMO
  560        OPEN        1      4        #LP3         MEMO
```

The LIST command creates job 560 and leaves job 534 unchanged. The C option preserves form-control information for job 560, such as overprinting of lines for boldface and underscores.

Using Peruse With TAL

The following example shows several features of Peruse that are useful for programmers. This example uses TAL, but also applies to other high-level languages, such as COBOL85 or Pascal.

Compiling Your Job

Run your compilation; include the NOWAIT option so that the compilation runs concurrently with your Peruse session:

```
1> TAL /IN SRCFILE, OUT $S.#HOLD, NOWAIT/ OBJFILE
```

Start Peruse and display your job. Peruse displays the status of your job as OPEN because the TAL compiler is still running:

```
2> PERUSE
PERUSE - T9101D10 - (08JUN92)  SYSTEM  \WEST
Copyright Tandem Computers Incorporated 1978, 1982, 1983, 1984, 1985, 1986,
1987, 1988, 1989, 1990, 1991

  JOB  BATCH STATE PAGES COPIES PRI HOLD LOCATION     REPORT
  135        OPEN        1      4        #HOLD        TAL USER
```

Monitoring the Job Status

Use the STATUS command to monitor your job while it is open. The STATUS command displays a constantly updated list of your jobs and notifies you with a beep when the compilation is finished. To redisplay the Peruse prompt at this point, press Break:

```
_STATUS
  JOB  BATCH STATE PAGES COPIES PRI HOLD LOCATION      REPORT
J 135       OPEN        1      4   B    #HOLD         TAL USER
(BEEP)
  JOB  BATCH STATE PAGES COPIES PRI HOLD LOCATION      REPORT
JC135       READY 130   1      4        #HOLD         TAL USER
(BREAK key)
```

Finding TAL Errors

Instead of waiting for your TAL compilation to finish, you can use the FIND, PAGE, and LIST commands to search for errors. Because all TAL compilation errors begin with the string **** ERROR, you can use this string to search for errors in your program. For example:

```
_FIND /**** ERROR/
**** ERROR 27 **** Illegal syntax
```

Continue using the FIND command to find any remaining errors; if you don't specify a new string, FIND continues to search for the same string. Determine the page on which each error occurs with the PAGE command and then display the page, with the LIST command:

```
_FIND
**** ERROR 54 **** Illegal reference parameter
_PAGE
PAGE: 8      LINE: 12
_LIST *
```

Continue searching for the error string until there are no more occurrences of it in the program. For example:

```
_FIND
**** ERROR 72 **** Indirection must be supplied
_FIND
```

When you have found all the errors in the program and are through with this TAL listing, delete it with the DEL command and exit Peruse:

```
_DEL ; EXIT
```

Correct the errors in your source file and run TAL again.

The following example shows a TAL compilation with the compiler listing sent to location $S.#HOLD:

```
11> TAL /IN PROG, OUT $S.#LOOK/ TALOBJ
12> PERUSE
PERUSE - T9101D10 - (08JUN92)    SYSTEM  \WEST
Copyright Tandem Computers Incorporated 1978, 1982, 1983, 1984, 1985, 1986,
1987, 1988, 1989, 1990, 1991

  JOB  BATCH STATE PAGES COPIES PRI HOLD LOCATION     REPORT
  136        READY 130   1      4        #HOLD        TAL USER
```

Use the LIST LAST (or LL) command to display the last page of the listing, which is the TAL trailer message. This message indicates that no warning or error messages were generated during the compilation:

```
_LIST LAST
Object file name is $VOL1.ADMIN.PROG
This object file will run on either a TNS or a TNS/II system
Number of errors = 0
Number of warnings = 0
Primary global storage = 44
Secondary global storage = 1120
Code size = 1888
Data area size = 40 pages
Code area size = 2 pages
Maximum symbol table space available = 24892, used = 4064
Maximum extended symbol table space available = 0, used = 0
Number of source lines = 3224
Elapsed time - 00:11:05
```

To print this job, use the LOC command to send it to a location associated with an actual printer. When your job prints, Peruse displays the status as PRINT:

```
_LOC #LP; JOB

  JOB  BATCH STATE PAGES COPIES PRI HOLD LOCATION     REPORT
  136        PRINT 130   1      4        #LP          TAL USER
_EXIT
```

Using Peruse With Files

In addition to spooler jobs, you can also use Peruse to manage spooler job files and EDIT files. A spooler job file is an unstructured disk file with file code 129. This type of file contains a spooler job, including print data records and formatting control information. Two uses for spooler job files are:

- Formatted documents

  A document that has been formatted by a text formatter such as TFORM can be stored in a spooler job file and then printed without having to rerun the formatter.

- Compiler listings

  A compiler listing stored in a spooler job file can be printed without having to recompile the source code.

You can use these Peruse commands with spooler job files:

```
FILES       LIST        PURGE       STATUS
FIND        NUMCOL      SJFILES     VOLUME
JOB         PAGE        STARTCOL
```

Peruse commands that cannot be used with spooler job files are COPIES, DEL, FORM, HOLD, HOLDAFTER, LOC, PRI, and REPORT. If you issue one of these commands for a spooler job file, Peruse displays the following message:

```
COMMAND NOT SUPPORTED FOR SPOOLER JOB FILES
```

To let you access spooler job files, Peruse can run without the supervisor process. If the supervisor process is not running and you enter a command that requires this process, Peruse displays the message:

```
ACCESS TO SPOOL SUPERVISOR PROCESS FAILED, FILE ERROR 016
```

Report this message to your system manager or operator. The supervisor process must be running before you can execute the command.

Copying a Spooler Job to a Spooler Job File

To copy a spooler job generated by the COBOL85 compiler to a spooler job file:

1. Generate the spooler job with the COBOL85 compiler and send the job to a holding location to prevent the job from printing:

```
9> COBOL85 / IN SRCFILE, OUT $S.#HOLD / COBJECT
```

2. Start Peruse and examine your jobs on the queue:

```
10> PERUSE
PERUSE - T9101D10 - (15FEB89)    SYSTEM  \WEST
Copyright Tandem Computers Incorporated 1978, 1982, 1983, 1984, 1985, 1986,
1987, 1988, 1989, 1990, 1991

  JOB  BATCH STATE PAGES COPIES PRI HOLD LOCATION      REPORT
  97         READY 2     1      4        #LAZR         LETTER
  150        READY 16    1      4        #HOLD         USERS COBOL
```

   Job number 97 is already on the queue waiting to print. The COBOL85 listing is job number 150, which is the current job because it is the most recently spooled job on the queue.

3. Use the LIST command to copy the job from the spooler to the spooler job file named LISTFILE in your current subvolume:

```
_LIST COMPRESS /OUT LISTFILE !/ ALL
```

   Peruse copies job 150 to the LISTFILE file. This file now contains the COBOL85 listing and formatting control information.

   The exclamation point (!) purges the LISTFILE file (if it exists) before the copy begins. If you do not specify the exclamation point and LISTFILE already exists, Peruse appends the spooler job to the file.

   The LIST command in this example includes the COMPRESS keyword, which specifies ASCII compression for the spooler job file. ASCII compression saves disk space by compressing 8-bit ASCII characters into five bits each in the spooler job file.

4. If you know that job 150 is your most recently spooled job, you can also enter the above commands from your TACL prompt:

```
10> PERUSE ; LIST COMPRESS / OUT LISTFILE ! / ALL
```

   To copy two or more spooler jobs to the same spooler job file, issue a LIST command for each spooler job, but include the exclamation point for only the first LIST command. The second and subsequent spooler jobs are appended to the spooler job file. This procedure is useful for concatenating several reports or listings into a single spooler job file.

Copying a Spooler Job to an EDIT File

1. Start Peruse and examine your jobs on the queue:

```
10> PERUSE
PERUSE - T9101D10 - (08JUN92)    SYSTEM  \WEST
Copyright Tandem Computers Incorporated 1978, 1982, 1983, 1984, 1985, 1986,
1987, 1988, 1989, 1990, 1991

  JOB  BATCH STATE PAGES COPIES PRI HOLD LOCATION      REPORT
  97         READY 2     1      4        #LAZR         LETTER
  150        READY 16    1      4        #HOLD         USERS COBOL
```

2. Copy job number 97 to an edit file.

   a. Execute a JOB command to make it the current job.

   b. Use the LIST command with the EDIT option to copy job number 97 from the spooler to the edit file named LETTER in your current subvolume:

```
_JOB 97 ; LIST EDIT /OUT LETTER !/ ALL
```

   Peruse copies job 97 to the file LETTER. The exclamation point (!) purges the file (if it exists) before the copy begins. If you do not specify the exclamation point and LETTER already exists, Peruse appends the spooler job to the file. If the file LETTER does not exist, Peruse creates it.

3. If you know the job number, you can enter the above commands from the TACL prompt:

```
10> PERUSE; JOB 97 ; LIST EDIT /OUT LETTER ! / ALL
```

Copying a Spooler Job File to the Spooler

To copy a spooler job file from your current subvolume to a spooler job:

1. Start Peruse and use the SJFILES command to display the spooler job files in your subvolume:

```
10> PERUSE
PERUSE - T9101D10 - (08JUN92)    SYSTEM  \WEST
Copyright Tandem Computers Incorporated 1978, 1982, 1983, 1984, 1985, 1986,
1987, 1988, 1989, 1990, 1991

_SJFILES
\WEST.$DISK2.USERS
  .REPORTL
```

2. Use the JOB command to make REPORTL the current job.

3. Use the LIST command to copy REPORTL to the spooler queue location $S.#LAZR. The original REPORTL file remains unchanged in your subvolume.

```
_JOB REPORTL
_LIST /OUT $S.#LAZR/ ALL
```

You can also enter the above commands from the TACL prompt:

```
10> PERUSE ; JOB REPORTL ; LIST /OUT $S.#LAZR/ ALL
```

14

Performing Routine Spooler Operations Using Spoolcom

The spooler provides the interface between you (and your applications) and the printers on your system.You use the Spoolcom utility to manage the spooler.

This section describes some of the Spoolcom commands that are available for all users.

Some Spoolcom commands affect the entire spooler subsystem and therefore are restricted to super-group users (user ID 255,*n*), such as your system operator. For information about these commands, see the *Spooler Utilities Reference Manual*.

Entering Spoolcom Commands

You can enter Spoolcom commands using any of these methods:

Entering Individual Spoolcom Commands at the TACL Prompt

Eenter the keyword Spoolcom followed by the command. Spoolcom executes the command and returns you to the TACL program, which displays another TACL prompt. To display the status of the printer $LP1:

```
10> SPOOLCOM DEV $LP1

DEVICE                STATE          FLAGS   PROC    FORM
$LP1                  WAITING           T    $SPLA
```

The Spoolcom DEV command expects a printer device name, not a printer location name. Use the Spoolcom LOC command to list all printer locations on your node and the device name of each location.

Starting a Spoolcom Process and Entering Commands Interactively

1. Enter the keyword Spoolcom at the TACL prompt.

   Spoolcom displays its program banner, copyright message, and the Spoolcom prompt, which is a close parenthesis:

```
12> SPOOLCOM
SPOOLCOM - T9101D10 - (08JUN92)    SYSTEM  \WEST
Copyright Tandem Computers Incorporated 1978, 1982, 1983, 1984, 1985,
1986,
1987, 1988, 1989, 1990, 1991
)
```

   Fields in the Spoolcom banner are:

   T9101D10 is the product number and version for Spoolcom.

   (08JUN92) is the release date for this version of Spoolcom.

   \WEST is the system on which this Spoolcom process is running.

2. Enter a Spoolcom command at the prompt.

   After executing the command (or displaying an error message if the command is invalid), Spoolcom redisplays its prompt and waits for another command.

3. To terminate Spoolcom, enter the EXIT command or CTRL-Y.

Entering Commands From a Command File

Spoolcom accepts commands from a command file — an unstructured EDIT disk file that contains one or more commands, created with a text editor such as TEDIT.

When you start a Spoolcom process, specify the command file for the IN option. Spoolcom executes the commands in this file then returns control to the TACL program. For example, the command file DEVSTAT contains these Spoolcom commands:

```
COMMENT -- SPOOLCOM Device Status Command File
COMMENT  Displays the status of all line printers
DEV $LP1
DEV $LP2
DEV $LP2
```

To execute these commands, run Spoolcom and specify DEVSTAT for the IN option:

```
12> SPOOLCOM / IN DEVSTAT /
```

Spoolcom displays the status of $LP1, $LP2, and $LP3 and then returns control of your terminal to the TACL program.

Comment lines in the command file identify the file and explain the operations being performed. Each comment line must begin with the word COMMENT, which tells Spoolcom to ignore the rest of the characters on that line.

Spoolcom Commands

Spoolcom commands entered from a terminal or read from a command file cannot exceed 132 characters. You can enter two or more Spoolcom commands on the same line if you separate them with semicolons. For example, this command displays the status of job number 322, then exits Spoolcom:

```
)JOB 322, STATUS; EXIT

JOB  BATCH STA FLAGS OWNER  TIME COPY PAGE  REPORT    LOCATION
322        RDY    T   1,30  11:23 1    12   MANFJOE   #LP
```

Spoolcom commands consist of a command word sometimes accompanied by a parameter, and optionally followed by any number of subcommands. The command and its parameter are separated from the subcommands by commas; subcommands are separated from each other by commas.

For example, to specify the report name TAL COMPILE for job number 1635:

```
)JOB 1635, HOLD, REPORT TAL COMPILE, START
```

JOB	The command; parameter 1635 references job number 1635 for the command
HOLD	Subcommand that places the job in the hold state (to rename a job report, you must first put the job on hold)
REPORT	Subcommand that specifies the report name TAL COMPILE for the job
START	Subcommand that places the job back in the printer queue

If you enter commands on separate lines, each command must be complete. For example, to enter the subcommands from the previous example on three separate lines, you must repeat the command JOB and the parameter 1635 on each line:

```
)JOB 1635, HOLD
)JOB 1635, REPORT TAL COMPILE
)JOB 1635, START
```

Each subcommand can be considered a separate command. Subcommands are processed left to right, and each subcommand is processed before the next subcommand is executed. (The SPOOLER command with the DRAIN option is an exception. Spoolcom puts the component in the drain state but does not wait for the drain to complete.)

Spoolcom Command Summary

Not all subcommands for each command are listed below. For complete information about all Spoolcom commands, see the *Spooler Utilities Reference Manual*.

Table 14-1. Spoolcom Commands (Super-Group Users Only) (page 1 of 2)

Command	Subcommand	Function
BATCH		Specifies attributes, obtains status, changes batch job states.
COLLECT		Specifies attributes, obtains status, changes collector states.
	DELETE	Removes a specified collector from the spooler.
	DRAIN	Causes the collector to stop accepting new jobs.
	START	Causes a dormant collector to become active.
	STATUS	Shows current status of the collector processes.
COMMENT		Designates a comment for Spoolcom to ignore; often used in command files.
DEV		Specifies attributes, obtains status, changes device states.
	CLEAR	Stops printing the current job on a given device.
	DELETE	Removes a device from the spooler.
	DRAIN	Causes a device to go offline after the current job finishes.
	START	Causes an offline device to become online.
	STATUS	Shows current status of devices.
	SUSPEND	Causes a device to stop all activity.
	XREF	Produces a cross-reference list of devices, locations, and print processes, ordered by device.
EXIT		Terminates a Spoolcom session.
FC		Allows edit and reexecution of a command line.
FONT		Designates a special control job in the spooler as a font job, containing printing instructions for programmable printers like vertical form control (VFC) commands. Also displays status of spooler jobs associated with specific fonts.

Table 14-1. Spoolcom Commands (Super-Group Users Only) (page 2 of 2)

Command	Subcommand	Function
HELP		Displays the syntax of Spoolcom commands.
JOB		Specifies attributes, obtains status, and changes job states.
	DELETE	Deletes a job from the spooler.
	HOLD	Places a job in the hold state.
	HOLDAFTER	Sets the hold-after-printing flag on or off. HOLDAFTER or HOLDAFTER ON causes the job to be placed in the hold state after it finishes printing. HOLDAFTER OFF (the default setting) lets the job be deleted after printing.
	START	Takes a job out of the hold state, places it in the ready state.
	STATUS	Displays the status of all jobs or of a given job.
LOC		Defines, modifies, or displays status of, spooler routing structures; specifies the font job to download to a location.
	DELETE	Deletes from the spooler an entire group or a particular destination within a group.
	DEV	Connects or disconnects a location to a device.
	STATUS	Shows current status of the spooler locations.
	XREF	Produces a cross-reference list of locations, devices, and print processes ordered by location.
OPEN		Specifies the spooler supervisor with which Spoolcom communicates.
PRINT		Specifies attributes, obtains status, changes print process states.
	DELETE	Removes a print process from the spooler.
	START	Takes a print process out of the procerror state after the cause of a failure has been remedied.
	STATUS	Displays the attributes of print processes.
	XREF	Produces a cross-reference listing of print processes, devices, and locations.
SPOOLER		Starts, stops, and obtains the status of the spooler.
	DRAIN	Brings the spooler to an orderly halt after all currently printing or spooling jobs have finished.
	DUMP	Obtains a spooler memory dump while it is running.
	MGRACCESS	Enables or disables manager access for the spooler.
	START	Starts the spooler collectors and print processes.
	STATUS	Displays the status of the spooler subsystem.

Listing Printers and Checking Their Status

To check the status of all printers on your system, enter:

```
> SPOOLCOM DEV
```

A listing similar to this is sent to your home terminal:

```
DEVICE                          STATE          FLAGS  PROC    FORM
\SAGE.$S1                       WAITING        H      $SPLX
\SAGE.$S2                       WAITING        H      $SPLX
\AMBER.$S                       WAITING        H      $SPLP
\AMBER.$S2                         WAITING          H       $SPLX
```

WAITING in the STATE column indicates that the above printers are all available to print users' jobs. Table 14-2 describes the possible states for a printer.

Example

To check the status of the printer $LASER, enter:

```
> SPOOLCOM DEV $LASER
```

A listing such as this is sent to your home terminal:

```
DEVICE                          STATE          FLAGS  PROC    FORM
$LASER                          WAITING        H      $SPLP
```

WAITING in the STATE column of this listing shows that the printer $LASER is up and available to print users' jobs.

Table 14-2. Printer Device States

State	Definition
Busy	The device is currently printing a job.
Waiting	The device is ready to print a job, but no job is available to print on the device, either because there are no jobs in the device queue or because none of the jobs in the queue have a form name that matches the form currently on the device.
Offline	The device is not available for printing jobs. The Spoolcom command DEV, DRAIN causes the device to go offline after the current job finishes printing. Device attributes can be changed only when the device is offline.
Suspended	The device stopped all activity as a result of the DEV, SUSPEND command. If it was printing a job when it was suspended, that job will resume printing when the device is restarted.
Procerror	The supervisor determined that the device print process is not working correctly and has sent an error message. The device is unusable until the print process is restarted. See the *Operator Messages Manual* for information on spooler print process error messages.
Deverror	A file-system error occurred on the device while a job was printing. File-system error messages are listed in the *Operator Messages Manual* and are described more fully in the *Guardian Procedure Errors and Messages Manual*.

Restarting a Printer

If a printer is offline because of a device error, you can determine the error condition with Spoolcom. Spoolcom displays both the file-system error number (in this case, error 100, NOT READY) and the job numbers in the printer queue. In this example, device error 100 occurred for the $LP1 printer:

```
18> SPOOLCOM DEV $LP1

DEVICE                      STATE          FLAGS  PROC    FORM
$LP1                        DEV ERROR 100     T   $SPLP

JOB  LOCATION          DEVICE               SEQ   COPY  PAGE
571  #LP1.DEFAULT      $LP1                 1     1     40
596  #LP1.DEFAULT      $LP1                 2     1     43
```

You can use Spoolcom to restart a printer that has gone offline because of a device error. To restart $LP1:

1. Correct the device error (in this case, make the printer ready).
2. Issue a DEV command:

```
19> SPOOLCOM DEV $LP1, START
```

Table 14-3. Common Device Errors

Error	Meaning	Recovery
14	DEVICE DOES NOT EXIST	Check if the printer exists by entering: 20> PUP LISTDEV *device-name*
100	NOT READY	Put the printer online (press the READY button on the printer).
102	PAPER OUT	Reload the printer with paper and make it online.

You can also use the Peruse DEV command to examine the status of a printer and receive not only the device status, but the owner ID and the printing times for the queued jobs. In this example, the device state shows an error number for the printer $LP1, indicating that this printer is offline:

```
_DEV $LP1

DEV STATE: ERROR 100     FORM:
JOB   OWNER    PAGES  WAIT       FORM
571   008,013  40     00:02:32+
596   008,013  43     00:03:39+
_EXIT
```

A plus sign (+) following an estimated printing time indicates that the printer must come back online before the print time has any meaning.

Displaying the Status of Spooler Components

Spoolcom can display the status of collectors, printers, print processes, the spooler routing structure, and the spooler itself, as well as any spooler job you own.

Table 14-4. Spoolcom Commands for Displaying Spooler Component Status

Spooler Component	Spoolcom Command
Collector	COLLECT
Printer	DEV
Job	JOB
Batch Job	BATCH
Print Process	PRINT
Routing Structure	LOC
Spooler	SPOOLER

You can use these commands from a TACL or Spoolcom prompt, or a command file.

For example, to display the status of the spooler process, enter the Spoolcom SPOOLER command at the TACL prompt:

```
14> SPOOLCOM SPOOLER
SPOOLER           STATE      LOGGING FILE              LAST ERROR
$SPLS             ACTIVE     $0
```

Spoolcom displays the spooler process status and returns you to the TACL program.

Your spooler subsystem has many printers, jobs, and related routing structures. Therefore, to display only specific information, specify a single entity such as a job number, the name or number of a printer, or a location name:

1. Use Peruse to list your spooler jobs in the spooler system, and the job location.

   For example, this display shows that your job is number 566 at location #HT4:

```
  JOB  BATCH STATE PAGES COPIES PRI HOLD LOCATION   REPORT
  566        OPEN        1      4        #HT4       MKTG BOB
_
```

2. Exit Peruse, and use the Spoolcom LOC command to display the status of the printer that is to print your job.

   For example, to find the location assigned to printer #HT4 and the jobs in its queue:

```
15> SPOOLCOM LOC #HT4

LOCATION           FLAGS  DEVICE                    FONT
#HT4.DEFAULT              $HT4
JOB  LOCATION          DEVICE              SEQ   COPY  PAGE
364  #HT4.DEFAULT      $HT4                PRINT 1     6
513  #HT4.DEFAULT      $HT4                2     1     6
566  #HT4.DEFAULT      $HT4                3     1     4
```

Use the number sign (#) and not the dollar sign ($) with the LOC command. If you use the dollar sign, you get this error message:

```
16> SPOOLCOM LOC $HT4
OPEN $SPLS    ; LOC $HT4
                    ^
INVALID COMMAND PARAMETER
```

Monitoring Spooler Processes

You might routinely need to monitor the status of the spooler supervisor, collector, or print processes.

Monitoring the Spool Supervisor

To monitor your spooler supervisor, enter:

```
> SPOOLCOM
```

If you do not specify a supervisor name, Spoolcom assumes the supervisor is $SPLS.

If this produces the standard startup banner:

```
   SPOOLCOM - T9101D20 - (08JUN93)   SYSTEM  \system
```

the supervisor is running.

If this error message is added:

```
SPOOLCOM - T9101D20 - (08JUN93)    SYSTEM  \system

OPEN $supervisor-name
          ^
SPOOLER OPEN ERROR 14
```

the supervisor is not running. See Warmstarting a Drained Spooler on page 14-12.

Monitoring Spooler Collector Processes

Do not let the collector processes in your spooler subsystem become more than about 90 percent full. To check the status of your spooler collector processes, enter:

```
> SPOOLCOM COLLECT
```

A listing such as this is sent to your home terminal:

```
COLLECT  STATE        FLAGS  CPU   PRI  UNIT  DATA FILE                %FULL
$S       ACTIVE              0 , 1 149  4     $SPOOL.SPL.DATA          40
$S1      ACTIVE              1 , 2 149  10    $SPOOL.SPL.DATA1         28
$S2      ACTIVE              2 , 3 149  8     $SPOOL.SPL.DATA2          0
```

This example shows that the three collector processes, $S, $S1, and $S2, are active and none is approaching a full state. If the %FULL column shows any collector process approaching 90 percent capacity, you should delete jobs from the collector in question.

Table 14-5. Collector Process States

State	Definition
Dormant	The collector cannot accept new jobs for spooling and no jobs are currently being spooled. You can declare, initialize, or delete a collector, or set and modify its attributes while dormant. The Spoolcom command COLLECT, DRAIN makes an active collector dormant once it finishes spooling all currently open jobs.
Active	The collector can accept new jobs for spooling. You cannot change its attributes while it is in the active state. The Spoolcom command COLLECT, START makes a dormant collector active.
Drain	The collector does not accept new jobs for spooling, but currently spooling jobs continue until completion. When all open jobs are spooled and all currently printing jobs associated with the collector have finished, the collector enters the dormant state. You cannot change collector attributes while it is in the drain state.
Error	The collector cannot function. The Spoolcom COLLECT, STATUS command tells you whether the collector is in an error state.

Monitoring Spooler Print Processes

To check the status of your print processes, enter:

```
> SPOOLCOM PRINT
```

A listing such as this is sent to your home terminal:

```
PRINT      STATE      FLAGS   CPU      PRI
$SPLA      ACTIVE             3 , 2    128
$SPLB      ACTIVE             2 , 3    144
$SPLC      DORMANT            2 , 3    128
$SPLD      DORMANT            3 , 2    144
$SPLX      DORMANT            1 , 0    149
```

Table 14-6. Print Process States

State	Definition
Active	The print process is running for one of these reasons: • The print process is printing a job. • A device controlled by the print process has been declared exclusive and must therefore be kept open regardless of whether a job is printing. • The print process is independent and always running. For more information on independent print processes, see the *Spooler Programmer's Guide*.
Dormant	The print process is not running. The print process enters the dormant state whenever it has no job to print, controls only shared devices, and is not an independent print process.
Procerror	When the supervisor determines that a print process is not responding correctly, it places the print process into the procerror state, and you receive an error message. See the *Operator Messages Manual* for information on spooler print process error messages.

You can modify the attributes of a print process when it is in the dormant or procerror state. The spooler can be active, warm, or cold. See Section 15, Managing the Spooler Using Spoolcom, for instructions on modifying print process attributes.

Draining the Spooler

You must drain, or stop, the spooler before you restart it through a warmstart or a coldstart, if you cannot get jobs out of the spooler, or if an error has occurred with any of the spooler processes.

△ **Caution.** Do not issue a TACL STOP command for any spooler process unless a SPOOLER, DRAIN operation does not succeed: spooler recovery from STOP can be time-consuming. First try to bring the spooler to an orderly halt with the SPOOLER, DRAIN command. See Stopping the Spooler on page 14-17 to use the TACL STOP command to bring down your spooler.

To drain the spooler:

1. Make sure you are logged on as a super-group user (255,*n*).
2. Drain the spooler:

```
> SPOOLCOM SPOOLER, DRAIN
```

The spooler moves from the active to the dormant state. Once the supervisor stops, you must use the SPOOL program to restart the spooler. See Warmstarting a Drained Spooler on page 14-12 for instructions on using the SPOOL program.

How Does Draining Work?

Draining the spooler brings the spooler to an orderly halt after all jobs currently printing or spooling have finished. When the spooler is in the drain state:

- The collectors stop accepting new jobs and reject new opens with file-system error 66 (DEVICE DOWNED BY OPERATOR). When all open jobs have finished spooling and all currently printing jobs have finished printing, the collectors enter the dormant state. Any attempt to route a job to a stopped collector is rejected with file-system error 14 (DEVICE DOES NOT EXIST).
- The print processes finish printing the jobs in the device queues and then enter the dormant state.
- The supervisor updates its control files and stops. The spooler is now in the dormant state and is ready to be warmstarted.

Once drained, the spooler consists solely of a set of disk files: program files containing object code, data files containing spooled jobs, and control files containing the names and attributes of the components and jobs known to the spooler.

Because there is no supervisor with which Spoolcom can communicate, you cannot obtain information about a dormant spooler.

Starting a Drained Spooler

Warmstarting a spooler is theact of bringing the spooler from the warm state to the active state. When the spooler is dormant, the supervisor is not running. As soon as you create another supervisor process, the spooler enters the warm state. When you warmstart the spooler, you use the same control files and other files that were in use when the spooler was previously drained.

Coldstarting a spooler is the act of starting a new spooler. When you coldstart a spooler, you create new control files in which the collectors store jobs (when you warmstart the spooler, the collectors use existing control files).

The key distinction is that in a coldstart the supervisor is run with a new control file, while in a warmstart the supervisor is given the name of an existing control file. Before you coldstart your spooler, you must purge all existing supervisor control files. The coldstart process creates new control files. Any jobs waiting in the current queue file are purged during a coldstart, and users must respool these jobs after the coldstart finishes.

Warmstarting a Drained Spooler

1. Log on as a super-group user (255,*n*), but not as the super ID (255,255).
2. Run the spooler supervisor, supplying the name of an existing spooler control file:

   ```
   > SPOOL / IN control-filename , NAME $supervisor-process /
   ```

   See the *Spooler Utilities Reference Manual* for a description of the SPOOL program.
3. Optionally add, delete, or modify collectors, print processes, devices, and routing structures (see Section 15, Managing the Spooler Using Spoolcom).
4. Start the spooler:

   ```
   > SPOOLCOM SPOOLER, START
   ```

Note. If you are warmstarting the spooler after migrating from one system version to another, see the migration considerations in the *Spooler Utilities Reference Manual.*

Automation Example

This command file warmstarts a spooler. You can adapt it by substituting elements specific to your system, then invoke it with an OBEY command:

```
> OBEY $SYSTEM.SPLUTIL.WARMFILE
```

```
COMMENT -- THIS IS $SYSTEM.SPLUTIL.WARMFILE
COMMENT -- It restarts a previously existing spooler system
COMMENT -- First, create the supervisor process, supplying the name of the
COMMENT    control file of the spooler system being warmstarted:

SPOOL /IN $MKT.SPL.SPL,OUT $0,NAME $SPLS,PRI 147,CPU 0/1

COMMENT -- Add or modify any collectors and print processes (to delete a
print COMMENT    process, first delete the locs and devices associated
with it):

SPOOLCOM PRINT $OFFICE,FILE $WORK.ALL.BEST,PRI 145,CPU 3
SPOOLCOM LOC #DEFAULT.FAST
SPOOLCOM DEV $SLOW,PROCESS $USERP,SPEED 300,WIDTH 70

COMMENT -- Finally, start the spooler:

SPOOLCOM SPOOLER, START

COMMENT -- The spooler is now completely active and ready to
COMMENT -- spool and print jobs.
```

Coldstarting a Drained Spooler

Note. Coldstarting a spooler subsystem is an infrequent task that you perform primarily in emergency situations. Before performing a coldstart, make sure it is the appropriate operation.

The spooler coldstart procedure is generally the same each time you do it. You should maintain command files to save time when you start the spooler. Automation Example on page 14-13 has a sample command file you can adapt for your system.

To start a new spooler:

1. Log on as a super-group user (255,*n*), but not as the super ID (255,255).
2. If you are going to change the data file size, purge the collector data file:

   ```
   > PURGE data-filename
   ```

   This is not required. If you are not going to change the data file size, do not purge the collector data file.
3. Purge the supervisor control files.

   ```
   > PURGE supervisor-control-filename
   ```
4. Create the data files in which collectors store jobs:

   ```
   > FUP CREATE data-filename [ , create-param ]...
   ```

   Create one data file for each collector. See Guidelines on page 14-16 for information on determining the extent size of the data file.

5. Run the spooler supervisor:

   ```
   > SPOOL / IN control-filename , NAME $supervisor-process /
   ```

   See the *Spooler Utilities Reference Manual* for a complete description of the SPOOL utility and its options.

6. Enter Spoolcom:

   ```
   > SPOOLCOM
   ```

7. Specify the names and attributes of the collectors:

   ```
   ) COLLECT $collector-name, DATA data-filename
   ```

   See Guidelines on page 14-16 for information on sizing the data file and for suggestions on the number of collectors you might need.

8. Specify the names and attributes of the print processes:

   ```
   ) PRINT $print-process
   ```

   Several PRINT subcommands are described in Table 14-1.

9. Specify the names and attributes of the devices:

   ```
   ) DEV $device , PROCESS $print-process
   ```

   Several DEV subcommands are described in Table 14-1.

10. Specify locations and connect locations to devices:

    ```
    ) LOC [ #group].dest , DEV [ $device ]
    ```

    The Spoolcom LOC command is described in Table 14-1.

11. Start the spooler:

    ```
    ) SPOOLER, START
    ```

Automation Example

These two command files coldstart a spooler in response to a single OBEY command. You can use these files as a model in setting up your own spooler coldstart command files. After you have entered these files into two separate EDIT files, enter:

```
> OBEY $SYSTEM.SPLUTIL.COLDFILE
```

The commands contained in this file are executed, and this text is displayed on your home terminal:

```
COMMENT -- THIS IS $SYSTEM.SPLUTIL.COLDFILE
COMMENT -- THIS COMMAND FILE CREATES A SPOOLER SYSTEM
COMMENT -- Purge any existing collector data files having the
COMMENT    same name as the data file you intend to use.
PURGE $MKT.SPL.DATAFILE
COMMENT -- Purge spooler supervisor control files having
COMMENT    names you intend to use:
PURGE $MKT.SPL.SPL0
PURGE $MKT.SPL.SPL1
PURGE $MKT.SPL.SPL2
PURGE $MKT.SPL.SPL3
PURGE $MKT.SPL.SPL4
PURGE $MKT.SPL.SPL5
PURGE $MKT.SPL.SPL6
PURGE $MKT.SPL.SPL7
PURGE $MKT.SPL.SPL8
PURGE $MKT.SPL.SPL9

COMMENT -- Create a new collector data file:
FUP CREATE $MKT.SPL.DATAFILE, EXT ( 16384, 0 )
COMMENT -- Create the spooler supervisor process,
COMMENT    specifying all coldstart parameters:

SPOOL / IN $MKT.SPL.SPL, OUT $0, NAME $SPLS, NOWAIT, &
        PRI 147, CPU 0/1, 8191, 4095, 511, 10, 10, 40, 500

COMMENT -- supervisor control file = $MKT.SPL.SPL[0-9]
COMMENT -- log file for messages   = $0 (op. console)
COMMENT -- supervisor process name = $SPLS
COMMENT -- execution priority      = 147
COMMENT    should be lower than editor
COMMENT    or interactive utilities
COMMENT -- primary CPU                      = 0
COMMENT -- backup CPU                       = 1
COMMENT -- maximum number of jobs           = 8191
COMMENT -- maximum number of locations      = 4095
COMMENT -- maximum number of devices        = 511
COMMENT -- maximum number of collectors     = 10
COMMENT -- maximum number of print processes= 10
COMMENT -- maximum number of font jobs      = 40
COMMENT -- maximum number of batch jobs     = 500
COMMENT -- The spooler is now in the cold state; the
COMMENT    supervisor is running.  The next command runs
COMMENT    SPOOLCOM, specifying a spooler "configuration
COMMENT    file" to initialize the spooler:

SPOOLCOM / IN $SYSTEM.SPLUTIL.SPLCONF /

COMMENT -- Start the spooler:

SPOOLER, START
```

The coldstart file above causes Spoolcom to execute the commands contained in this spooler configuration file named SPLCONF:

```
COMMENT -- THIS IS $SYSTEM.SPLUTIL.SPLCONF

COMMENT -- THIS CONTROL FILE USES SPOOLCOM TO CONFIGURE THE
COMMENT    SYSTEM TO BE COLDSTARTED AND PASSES THIS
COMMENT    INFORMATION TO THE SUPERVISOR.

COMMENT  *****  YOUR SYSTEM
COMMENT  *****  SPOOLER CONFIGURATION
COMMENT  *****  28 AUG 1993

COMMENT -- To configure one collector named "$S":

COLLECT $S, FILE $SYSTEM.SYSTEM.CSPOOL
COLLECT $S, DATA $MKT.SPL.DATAFILE
COLLECT $S, CPU 1, BACKUP 0, PRI 146

COMMENT -- To configure four print processes,
COMMENT    one PSPOOL, one PSPOOLB, one FASTP, and one
COMMENT    user-written print process:

PRINT $STAND, FILE $SYSTEM.SYSTEM.PSPOOL, PRI 145
PRINT $STAND, CPU 0, BACKUP 1

PRINT $PSPB, FILE $SYSTEM.SYSTEM.PSPOOLB, PRI 145
PRINT $PSPB, CPU 0, BACKUP 1

PRINT $FSTP, FILE $SYSTEM.SYSTEM.FASTP, PRI 145
PRINT $FSTB, CPU 0, BACKUP 1

PRINT $USERP, FILE $USER.USER.USER, PRI 145, CPU 2

COMMENT -- To configure two line printers:

DEV $FAST, PROCESS $STAND, SPEED 900
DEV $SLOW, PROCESS $USERP, SPEED 300, WIDTH 120

COMMENT -- To configure one group, consisting of two
COMMENT    locations:

LOC #DEFAULT.FAST, DEV $FAST
LOC #DEFAULT.SLOW, DEV $SLOW

EXIT
```

Guidelines

- Collector data files usually last for the life of the system. To prevent disk fragmentation, allocate the data files to an empty volume immediately after labeling the volume.
- When you configure the collector (Step 6 of the coldstart procedure), use the collector attribute:

  UNIT *unit-size*

  to specify the amount of space allocated for each unit needed by the collector. A unit is a 512-word block. *unit-size* multiplied by 1024 is the number of bytes

allocated each time a unit is needed. *unit-size* can be any integer from 2 through 32,767. One collector can handle a maximum of 8192 units.

The maximum logical data file size is:

```
(unit-size * 1024) * 8192 = maximum-data-file-size
```

where *maximum-data-file-size* is a number in bytes.

Determine the value of *unit-size* carefully, because you cannot increase its size after the collector is configured. If the value of *unit-size* is large and the majority of spooled jobs are small, the last allocated unit of each job can contain a lot of unused space. On the other hand, if the value of *unit-size* is small and the majority of spooled jobs are large, more units must be allocated for each job and more time is required to complete the jobs.

- Once you calculate the unit size needed for the data file, you can determine the data file's total extent size.

  The data file can have up to 16 extents. However, to prevent disk space fragmentation, allocate the collector data file in one extent. You specify the extent size when you create the collector data file (Step 3 of the coldstart procedure). For example, if *unit-size* is 4, the extent size is 16384 and you create the data file with the command:

  ```
  > FUP CREATE file-name, EXT ( 16384, 0 )
  ```

- If the size of spooled jobs varies greatly, configure several collectors with different unit sizes. For example, configure $SS with unit size 2 for small jobs, $SM with unit size 4 for medium jobs, and $SL with unit size 8 for large jobs.
- You cannot change the data file size once you start the collector. To change the data file size:

  1. Drain the spooler.
  2. Create a larger file and change the unit size if necessary.
  3. Coldstart the spooler.

- File-system error 45 might indicate that no more units are available for a job because the file is full.

Stopping the Spooler

Stopping the spooler subsystem is similar to draining it. When you enter the SPOOLER, DRAIN command, the spooler should come to an orderly halt after all currently printing or spooling jobs finish.

If a SPOOLER, DRAIN operation does not succeed (that is, spooler activity does not stop), you must stop the collectors, print processes, and spooler supervisor with the TACL STOP command.

You might need to use the TACL STOP command if you cannot get jobs out of the spooler, if the supervisor isn't running, or if an error has occurred with a print process.

△ **Caution.** Do not use the TACL STOP command as a regular way of bringing down the spooler subsystem: recovery from the STOP command can be time-consuming. You should first try to bring the spooler to an orderly halt with the Spoolcom command SPOOLER, DRAIN. Use the TACL STOP command only after the SPOOLER, DRAIN command fails to work.

1. To stop your spooler subsystem with the TACL STOP command, exit from Spoolcom and enter:

   ```
   > STOP $supervisor
   > STOP $collector
   > STOP $print-process
   ```

2. Warmstart the spooler (use your warmstart command file):

   ```
   > OBEY warmstart-file
   ```

   See Warmstarting a Drained Spooler on page 14-12 for an example of a command file you can adapt and use for your system.

3. Check the status of the supervisor, collectors, and print processes:

   ```
   > SPOOLCOM
   ) COLLECT
   ) PRINT
   ```

   If the status displays show that any processes are stopped or if they continue to receive error messages, see Solving Common Spooler Problems on page 14-26.

Example: Stopping and Restarting a Spooler Subsystem

In this example:

- The supervisor is $SPLS.
- The collectors are $S, $S1, and $S2.
- The print processes are $FPLP, $SPL2, $SPLP, and $SPLX.

To stop, warmstart, and check the status of the spooler:

1. Exit Spoolcom:

   ```
   ) EXIT
   ```

2. Stop the supervisor:

   ```
   > STOP $SPLS
   ```

3. Stop the collector processes:

```
> STOP $S
> STOP $S1
> STOP $S2
```

4. Stop the print processes:

```
> STOP $FPLP
> STOP $SPL2
> STOP $SPLP
> STOP $SPLX
```

5. Warmstart your spooler subsystem using your command file named WARMFILE:

```
> OBEY WARMFILE
```

6. Check the status of the supervisor:

```
> SPOOLCOM
```

   It is running if the standard Spoolcom banner appears.

7. Check the status of the collectors:

```
) COLLECT
```

   If the collectors are in the active state, they are back up and running:

8. Check the status of the print processes:

```
) PRINT
```

   If the print processes are in the active or dormant state, they are back up and running. If any processes remain stopped or continue to receive error messages, see Solving Common Spooler Problems on page 14-26.

Controlling Print Devices

This subsection explains how to monitor, start, and stop the output devices associated with your spooler subsystem.

- To monitor your print devices, enter:

```
> SPOOLCOM DEV
```

- To monitor the status of a specific print device, enter:

```
> SPOOLCOM DEV $device
```

See Listing Printers and Checking Their Status on page 14-6 for more information about using the Spoolcom DEV command.

- To start a print device, enter:

```
) DEV $device, START
```

   This command causes an offline device to become online.

- To stop a print device, enter:

  ```
  ) DEV $device, SUSPEND
  ```

  This command causes the device to stop all activity. If the device is printing a job, the same job resumes printing when you restart the device.

Example

To check the status of a print device named $LASER, stop it with the DEV, SUSPEND command, restart it with the DEV, START command, and check its status one more time, enter:

```
> SPOOLCOM
) DEV $LASER
) DEV $LASER, SUSPEND
) DEV $LASER, START
) DEV $LASER
```

A listing such as this is sent to your home terminal:

```
DEVICE                          STATE          FLAGS  PROC    FORM
$LASER                          WAITING        H      $SPLP
```

WAITING in the STATE column of this listing shows that the printer $LASER is up and available to print users' jobs.

Controlling Jobs

To manage your spooler jobs, you can use the Spoolcom JOB commands the same way you would use Peruse commands.

All users can perform these operations on their own jobs; you must be a member of the super group to change any attributes of a job belonging to another user.

- To monitor all current jobs in your spooler subsystem, enter:

  ```
  > SPOOLCOM JOB
  ```

- To check the status of a specific job in your spooler subsystem, enter:

  ```
  ) JOB job-number, STATUS
  ```

- To place a job on hold, enter:

  ```
  ) JOB job-number, HOLD
  ```

- To remove a job from the hold state, enter:

  ```
  ) JOB job-number, START
  ```

- To make a job print after or ahead of other jobs, enter:

  ```
  ) JOB job-number, SELPRI selection-priority
  ```

A job must be in the hold state to change its priority. When jobs are added to the spooler, they are given a default priority of 4; the range is 0 through 7. The higher the priority, the sooner the job prints.

- To reroute a single job, enter:

  ```
  ) JOB job-number, LOC #location
  ```

  A job must be in the hold state to change its location. If you omit #*location*, #DEFAULT is used.

- To reroute all jobs that are queued on a down device:

  1. Determine all the locations that are associated with the down device:

     ```
     ) DEV $device, XREF
     ```

  2. Enter a separate LOC, DEV command for each location that is connected to the device. This routes each location to a different device:

     ```
     ) LOC #group.dest, DEV $different-device
     ```

- To delete a job, enter:

  ```
  ) JOB job-number, DELETE
  ```

  The Spoolcom JOB command includes other subcommands and qualifiers. See the *Spooler Utilities Reference Manual* for a complete description of the JOB command.

- You can enable or disable manager access for the spooler so that a group manager (*n*,255) can list and access all jobs that belong to group members. The default setting for MGRACCESS is OFF. A MGRACCESS ON setting does not persist through a warmstart. If spooler users expect manager access to be enabled, you must explicitly set MGRACCESS ON after warmstarting the spooler.

- To determine whether manager access is enabled, enter:

  ```
  > SPOOLCOM SPOOLER, STATUS DETAIL
  ```

- To enable manager access, enter:

  ```
  > SPOOLCOM SPOOLER, MGRACCESS ON
  ```

- To disable manager access, enter:

  ```
  > SPOOLCOM SPOOLER, MGRACCESS OFF
  ```

Examples

- When you enter a Spoolcom JOB command, a listing similar to this is displayed on your home terminal:

```
JOB  BATCH STA FLAGS OWNER   TIME  COPY PAGE  REPORT        LOCATION

22         RDY 4     30,1    08/09 1    1     NET     MAIL    #TLOGOUT
23         RDY 4     30,1    08/09 1    6     NET     MAIL    #TLOGALL
24         RDY 4     30,1    08/09 1    4     NET     MAIL    #TLOG
35         OPN 4     30,1    08/09 1    0     NET     MAIL    #TLOGOUT
36         OPN 4     30,1    08/09 1    0     NET     MAIL    #TLOGALL
37          OPN 4      30,1     08/09 1     0      NET      MAIL     #TLOG
```

  See Job States on page 14-23 for an explanation of the possible entries in the STA column of this listing.

- To check the status of job number 37, enter:

  ```
  ) JOB 37, STATUS
  ```

- To place job 37 on hold, enter:

  ```
  ) JOB 37, HOLD
  ```

- To remove job 37 from the hold state, enter:

  ```
  ) JOB 37, START
  ```

- To make job 37 print ahead of other jobs by changing its selection priority, enter:

  ```
  ) JOB 37, HOLD
  ) JOB 37, SELPRI 7
  ```

- To change the location of job 37 to #TLOGOUT, enter:

  ```
  ) JOB 37, HOLD
  ) JOB 37, LOC #TLOGOUT
  ```

- To reroute job 37 (and any other jobs) from a down device:

  1. Determine all the locations that are associated with the down device:

     ```
     ) DEV \SAGE.$C, XREF
     ```

     A listing similar to this is displayed on your terminal:

```
DEVICE                                LOCATION            PRINT PROCESS

\SAGE.$C                            #BIRD6.DEFAULT      $SPLX
\SAGE.$C                            #EBIRD.DEFAULT      $SPLX
\SAGE.$C                            #EBIRD1.DEFAULT     $SPLX
\SAGE.$C                             #TAF3.DEFAULT      $SPLX
```

2. Enter a separate LOC, DEV command for each location connected to the device:

```
) LOC #BIRD6.DEFAULT, DEV \ROSE.$C
) LOC #EBIRD.DEFAULT, DEV \ROSE.$C
) LOC #EBIRD1.DEFAULT, DEV \ROSE.$C
) LOC #TAF3.DEFAULT, DEV \ROSE.$C
```

- To delete job 37, enter:

```
) JOB 37, DELETE
```

- To determine whether manager access is enabled for your spooler, enter:

```
) SPOOLER, STATUS DETAIL
```

```
SPOOL SUPERVISOR: $SPLS

    STATE: ACTIVE
    MGRACCESS: OFF
    LOG FILE: $0
    LAST ERROR: NONE
```

- To enable manager access, enter:

```
) SPOOLER, MGRACCESS ON
```

- To check the status of manager access again, enter:

```
) SPOOLER, STATUS DETAIL
```

```
SPOOL SUPERVISOR: $SPLS

    STATE: ACTIVE
    MGRACCESS: ON
    LOG FILE: $0
    LAST ERROR: NONE
```

- To change the location of job 566 to #HT5 and specify that two copies be printed:

```
17> SPOOLCOM JOB 566, HOLD, LOC #HT5, COPIES 2, START
```

Job States

A job can be in any of these states:

Open (OPN)	The job has been added to the spooler. It remains in this state until the collector has finished storing the data on disk.
Ready (RDY)	The job is ready to print, but has not yet begun to print because another job is ahead of it in the device queue or its location is not connected to a device.
Hold (HLD)	The job has been placed on hold in order to prevent it from printing or in order to change its attributes.
Print (PRT)	The job is being printed.

Controlling Locations

A location is the logical destination of a job that has been sent to the spooler subsystem. If a print device is associated with a location, that device becomes the job's physical location. Locations are always two-part names taking the form #*group*.*dest*, where #*group* is a group name, such as #LP, and.*dest* is a destination name, such as.LASER.

To control locations in your spooler subsystem:

1. Make sure you are logged on as a super-group user (255,*n*).
2. List the locations in your spooler subsystem:

   ```
   > SPOOLCOM LOC
   ```

3. Modify your spooler subsystem locations as needed:
   - To assign and connect a location to a print device:

     ```
     ) LOC #group.dest, DEV $device-name
     ```

   - To break the connection between a location and any devices with which it is associated:

     ```
     ) LOC #group.dest, DEV
     ```

   - To delete a location from the spooler:

     ```
     ) LOC #group.dest, DELETE
     ```

Examples: Controlling Locations

Listing All Locations Associated With Your Spooler

To list all the locations associated with your spooler, enter:

```
> SPOOLCOM LOC
```

A listing such as this is sent to your home terminal:

```
LOCATION           FLAGS  DEVICE                                  FONT
#BIRD1.DEFAULT          $SAGE.#BOOK1

#BIRD2.DEFAULT          \TAF.$C

#BIRDK.DEFAULT          \KTTY.$C

#BIRDS.DEFAULT          $LOST.#BOOK
```

LOCATION	#*group.dest* of the location whose status is being displayed.
FLAGS	Displays a B if broadcast is on.
DEVICE	Device associated with that location, if any.
FONT	Defines a special-control job in the spooler.

Assigning and Connecting a New Location

To assign and connect the new location #BIRDB.DEFAULT to the print device $SAGE.#BOOK2, enter:

```
) LOC #BIRDB.DEFAULT, DEV $SAGE.#BOOK2
```

Breaking the Connection Between a Location and Devices

To break the connection between the location #BIRDB.DEFAULT and any device with which it is associated, enter:

```
) LOC #BIRDB.DEFAULT, DEV
```

Deleting a Location

For the location #BIRD1.DEFAULT:

1. Delete any current jobs from #BIRD1.DEFAULT.
2. Delete the location:

```
) LOC #BIRD1.DEFAULT, DELETE
```

Solving Common Spooler Problems

There are several common spooler problems that you might need to periodically solve:

Freeing a Hung Spooler: Cannot Get Jobs In or Out

If you cannot get jobs into or out of the spooler, the spooler is probably hung. To restore the spooler to normal operation with minimum impact on the rest of the system:

1. Check the status of the supervisor:

   ```
   > SPOOLCOM
   ```

   If this produces the standard startup banner:

   ```
   SPOOLCOM - T9101D20 - (08JUN93)   SYSTEM  \system
   ```

   the supervisor is still running. Continue with Step 2.

   If this error message is added:

   ```
   SPOOLCOM - T9101D20 - (08JUN93)   SYSTEM  \system
   OPEN $supervisor-name
             ^
   SPOOLER OPEN ERROR 14
   ```

   the supervisor is no longer running because of file-system error 14 (DEVICE DOES NOT EXIST).

   a. Check for error messages that might indicate what happened to the spooler process.

   b. Use the TACL PPD command to check whether any collectors or print processes are still running:

      ```
      > PPD $collector
      > PPD $print-process
      ```

c. Stop any running collector or print processes with the TACL STOP command:

```
> STOP $collector
> STOP $print-process
```

Go to Step 3.

2. Drain the spooler:

```
) SPOOLER, DRAIN
```

If the drain doesn't work (that is, spooler activity doesn't stop), exit from Spoolcom and stop the collectors, print process, and spooler supervisor with the TACL STOP command:

```
) EXIT
> STOP $supervisor
> STOP $collector
> STOP $print-process
```

3. Warmstart the spooler (use your warmstart command file):

```
> OBEY warmstart-file
```

4. Check the status of the collectors:

```
> SPOOLCOM
) COLLECT
```

Check the %FULL column in the collector status listing to see how full the collectors are. If %FULL is close to 100%, there might not be room for new jobs. Delete or print some jobs. With regular observation, you can monitor the collector and take preventive steps to keep it below full capacity.

If a collector is in the error state, the octal number that follows the error (in the ERROR column of the listing) indicates the error condition. The error number and its explanation appear on the operator console.

Try to restart the collector:

```
) COLLECT $collector-name, START
```

5. If the collector doesn't start, drain the spooler and warmstart it again. If that fails, coldstart it. See Warmstarting a Drained Spooler on page 14-12, and Coldstarting a Drained Spooler on page 14-13 for instructions.

Example

To use the procedure described above to free a hung spooler:

1. Check the status of the supervisor:

```
> SPOOLCOM
```

This startup banner tells you that the supervisor is still running.

```
   SPOOLCOM - T9101D20 - (08JUN93)   SYSTEM  \AMBER
```

2. Drain the spooler; then exit Spoolcom:

```
) SPOOLER, DRAIN
) EXIT
```

3. Warmstart the spooler (with a command file similar to the example in Warmstarting a Drained Spooler on page 14-12):

```
> OBEY $SYSTEM.SPLUTIL.WARMFILE
```

4. Check the collectors:

```
> SPOOLCOM COLLECT
```

Freeing a Hung Spooler: Jobs Do Not Print

If you cannot get jobs out of the spooler, a print process or a device might be offline. To determine the cause of jobs not printing:

1. Check the status of the supervisor:

```
> SPOOLCOM
```

   If the supervisor isn't running, check whether any collectors or print processes are still running (with the TACL PPD command) and stop any running spooler processes (with the TACL STOP command). Then warmstart the spooler.

2. Check the print processes:

```
) PRINT
```

   If the state of any print process is ERROR %*num*, the explanation for that error condition appears on the operator console. See Clearing a Print Process Error State on page 14-29 for instructions.

3. Check the devices:

```
) DEV
```

   If a device is offline, there might be a hardware error:

   a. Physically check the device. Fix the device if necessary.

   b. Drain the device:

```
) DEV $device-name, DRAIN
```

   c. Restart the device:

```
) DEV $device-name, START
```

4. Check the device:

```
) DEV $device-name
```

   If problems continue to occur, escalate this matter to your operations management.

Clearing a Print Process Error State

If a print process is in an error state, the devices it controls might also be in an error state. When you restart the print process, the system should automatically remove the devices from the error state and bring them up.

1. Make sure that the configuration is correct.

   If the configuration is correct, restart the print process:

   ```
   ) PRINT $print-process, START
   ```

   If the configuration is not correct, correct the configuration and then start the print process:

   ```
   ) PRINT $print-process
   ```

   ```
   ) PRINT $print-process, START
   ```

2. Check the status of the print process to make sure that it restarted properly:

   ```
   ) PRINT
   ```

3. Check the status of the devices to make sure that they are back up and running:

   ```
   ) DEV
   ```

Example

1. Check the status of the spooler supervisor $SPLS on the system \KONA:

   ```
   > SPOOLCOM
   ```

```
SPOOLCOM - T9101D20 - (08JUN93)   SYSTEM  \KONA
) SPOOLER
SPOOLER           STATE      LOGGING FILE       LAST ERROR
$SPLS             ACTIVE     $0
```

   The supervisor $SPLS is running.

   For additional information, enter:

   ```
   ) SPOOLER, STATISTICS
   ```

   A listing such as this is sent to your home terminal:

```
SPOOL SUPERVISOR: $SPLS
    JOBS                      : 1770
    OPEN JOBS                 : 3
    SUPERVISOR OPENERS        : 10
    DEVICE QUEUE NODES        : 423
    RECEIVE QUEUE ENTRIES     : 1
    JOBS CURRENTLY PRINTING   : 0
```

2. Check the print processes:

   ```
   ) PRINT
   ```

A listing such as this is sent to your home terminal:

```
PRINT    STATE             FLAGS  CPU    PRI
$SPLA    DORMANT           D      1 , 2  144
$SPLB    ERROR %100001            1 , 2  144
$SPLC    DORMANT           D      2 , 1  144
```

This listing shows that the print process $SPLB has an error. The device status command shows that the devices controlled by $SPLB are also in an error state:

```
) DEV
```

A listing such as this is sent to your home terminal:

```
DEVICE      STATE           FLAGS    PROC     FORM
$LPPR4      PROC ERROR      H        $SPLB
$LPPR5      PROC ERROR               $SPLB
$LPPR6      PROC ERROR      H        $SPLB
```

3. Restart $SPLB to clear the error:

```
) PRINT $SPLB, START
```

4. Check the status of the print processes to make sure that they are up and running:

```
) PRINT
```

A listing such as this is sent to your home terminal:

```
PRINT    STATE          FLAGS  CPU    PRI
$SPLA    DORMANT        D      1 , 2  144
$SPLB    ACTIVE                1 , 2  144
$SPLC    DORMANT        D      2 , 1  144
```

5. Check the status of the devices:

```
) DEV
```

A listing such as this is sent to your home terminal:

```
DEVICE     STATE           FLAGS    PROC     FORM
$LPPR4     JOB 103         H        $SPLB
$LPPR5     WAITING                  $SPLB
$LPPR6     WAITING         H        $SPLB
```

This DEV listing shows that all devices are now in a normal state.

Clearing a Nonprintable Job

If a “device error 100” error occurs, a user might have sent data to the spooler subsystem that a printer device is unable to handle. This procedure can help you to determine the owner of the job in question and to place the problem job on hold.

To clear a nonprintable job:

1. Check the physical status of the printer.

2. Check the logical status of the printer:

   ```
   > SPOOLCOM
   ) DEV $printer
   ```

   (Error 100 appears in the DEV listing for this device.)

3. Determine the current job number:

   ```
   ) DEV $printer, STATUS DETAIL
   ```

4. Determine the owner of the job:

   ```
   ) JOB job-number
   ```

   Exit from Spoolcom:

   ```
   ) EXIT
   ```

   Use the TACL USERS program to determine the user:

   ```
   > USERS group-id,user-id
   ```

   For more information about the USERS program, see the *TACL Reference Manual*.

5. Reenter Spoolcom and place the job on hold:

   ```
   > SPOOLCOM
   ) JOB job-number, HOLD
   ```

6. Drain the printer:

   ```
   ) DEV $printer, DRAIN
   ```

7. Start the printer:

   ```
   ) DEV $printer, START
   ```

8. Notify the user that the job is on hold and that it contains data the printer cannot handle.

Example

To clear a job that the printer named $LASER1 is unable to print:

1. Check the physical status of the printer $LASER1.
2. Log on as a super-group user (255,*n*) and check the logical status of the printer:

   ```
   ) DEV $LASER1
   ```

   A listing such as this is sent to your home terminal:

```
DEVICE                          STATE             FLAGS  PROC    FORM
$LASER1                            DEV ERROR 100        H     $SPLX
```

3. Determine the current print job:

   ```
   ) DEV $LASER1, STATUS DETAIL
   ```

A listing such as this is sent to your home terminal:

```
DEVICE: $LASER1
    STATE: DEVICE ERROR
    LAST ERROR: %004144
    EXCLUSIVE: OFF
    FIFO: OFF
    HEADER: ON
    TRUNCATION: OFF
    DRAINING: NO
    PRINTING JOB: 88
    PARM: -1 (%177777)
    PROCESS: $SPLX
    RETRY: 5
    TIMEOUT: 360
    SPEED: 100
    WIDTH: -1
    FORM:
    RESTART: OFF
    DEVRESET: OFF
    DEVTYPE:
    STARTFF: OFF
    ENDFF: OFF
    CHARMAP: NONE
    PREXLATE: OFF
    LUTOFVALUE: CRFFCR
    LUEOLVALUE: CRLF
    LUEOLWHEN: LT132
```

Look for the entry PRINTING JOB; in this example, the job number is 88.

4. Determine the owner of the job:

   ```
   ) JOB 88
   ```

   A listing such as this is sent to your home terminal:

```
JOB  BATCH STA FLAGS OWNER   TIME  COPY PAGE  REPORT         LOCATION
88       PRT 4    8,001  08:12  1    3    SALES   BONNIE  #DEFAULT

JOB  LOCATION          DEVICE                              SEQ   COPY  PAGE
88   #LASER1.DEFAULT   $LASER1                                PRINT 1     3
```

   If the owner is not apparent from this listing, exit from Spoolcom and learn more about the user through the TACL USERS program:

   ```
   ) EXIT
   > USERS 8,1
   ```

   The USERS program displays information such as this:

```
GROUP . USER        I.D. #        SECURITY         DEFAULT VOLUMEID

SALES . BONNIE     008,001          NUNU            $DATA1.BONNIEF
```

5. Reenter Spoolcom and place the job on hold:

   ```
   > SPOOLCOM
   ) Job 88, HOLD
   ```

6. Drain the printer:

   ```
   ) DEV $LASER1, DRAIN
   ```

7. Start the printer:

   ```
   ) DEV $LASER1, START
   ```

8. Notify the user SALES.BONNIE that the job is on hold and that it contains data the printer cannot handle.

Clearing a Paper Jam

A paper jam can cause a “device error 100” error to occur for a printer.

To clear a paper jam physically and programmatically:

1. Make sure you are logged on as a super-group user (255,*n*).
2. Check the physical status of the printer in question. Different printers vary, but many display an error light or number that can assist you in finding the jam.
3. Check the logical status of the printer:

   ```
   > SPOOLCOM
   ) DEV $printer
   ```

   DEV error 100 appears in the case of a paper jam.

4. Suspend the printer:

   ```
   ) DEV $printer, SUSPEND
   ```

5. Check the status of the printer:

   ```
   ) DEV $printer
   ```

   The STATE column should report SUSP.

6. Physically fix the paper jam.
7. Restart the printer from the selected page number of the job you suspended:

   ```
   ) DEV $printer, SKIP-page-number
   ) DEV $printer, START
   ```

   See the *Spooler Utilities Reference Manual* for more information about the DEV command and the DEV, SKIP subcommand.

Example: Logically Cleaing a Paper Jam

1. Check the physical status of the printer $LASER1.
2. Check the logical status of the printer:

   ```
   > SPOOLCOM
   ) DEV $LASER1
   ```

A listing such as this is sent to your home terminal:

```
DEVICE                          STATE              FLAGS  PROC    FORM
$LASER1                         DEV ERROR 100      H     $SPLX
```

DEV error 100 appears in the case of a paper jam.

3. Suspend the printer:

   ```
   ) DEV $LASER1, SUSPEND
   ```

4. Check the status of the printer:

   ```
   ) DEV $LASER1
   ```

   A listing such as the following is sent to your home terminal:

```
DEVICE                          STATE              FLAGS  PROC    FORM
$LASER1                          SUSP JOB 89         H     $SPLX
```

   Note that the STATE column reports SUSP.

5. Physically fix the paper jam.
6. Restart the printer from the selected page number of the job you suspended:

   ```
   ) DEV $LASER1, SKIP 4
   ) DEV $LASER1, START
   ```

Recovering From an Invalid (Job -1) State

If you find an invalid (Job -1) state in the STATE column of a DEV status listing:

1. Drain the device to take it offline:

   ```
   > SPOOLCOM
   ) DEV $device, DRAIN
   ```

2. Check the device status:

   ```
   ) DEV $device
   ```

   If the Job -1 condition has been cleared, you can start the device:

   ```
   ) DEV $device, START
   ```

   If the Job -1 condition persists, drain and warmstart the spooler.

Example: Restoring an Invalid Device

1. Check the device status.

   ```
   ) DEV
   ```

A listing such as this is sent to your home terminal:

```
DEVICE            STATE          FLAGS  PROC    FORM
\SF.$S1           WAITING        H      $SPLX
\AM.$S1           Job -1             H !T   $SPLA
```

The Job -1 in the STATE column indicates that device \AM.$S1 has fallen into an invalid state.

2. Clear the invalid state:

   ```
   ) DEV \AM.$S1, DRAIN
   ```

3. Check the device status:

   ```
   ) DEV
   ```

   A listing such as this is sent to your home terminal:

```
DEVICE            STATE          FLAGS  PROC    FORM
\SF.$S1           WAITING        H      $SPLX
\AM.$S1           OFFLINE        H !T   $SPLA
```

4. Put the device back online:

   ```
   ) DEV \AM.$S1, START
   ```

Diagnosing Unusual Problems

If you encounter other problems with your spooler subsystem, performing a memory dump of the supervisor process's memory could provide information for fixing these kinds of problems:

- Spooler response time is inexplicably slow.
- The job command fails to take effect (jobs hang).
- The supervisor sends error messages continuously.

Problems such as the above are unusual, but they might occur in large installations where development is performed. You perform the dump operation while the spooler is running; you need not drain the spooler or stop any spooler processes before performing this operation.

To recover from unusual problems such as those listed above, you might need to dump the memory of the supervisor process to a disk file for analysis. Check with your operations management before performing a supervisor memory dump to make certain that this is the appropriate action.

To perform a memory dump of the supervisor's memory, enter:

```
> SPOOLCOM
) SPOOLER, DUMP filename
```

where *filename* is the name of a disk file the spooler creates for the purpose of this operation.

The spooler subsystem creates a disk file, type 130, that can be used to analyze the problems you are experiencing.

Example

To dump the memory of the supervisor process to a disk file named DUMP07, enter:

```
> SPOOLCOM SPOOLER, DUMP DUMP07
```

Problem-Solving Summary

Table 14-7 summarizes the common spooler-related problems discussed in this subsection and lists possible causes and solutions for such problems.

Table 14-7. Common Printer and Spooler Problems (page 1 of 2)

Problem	Possible Causes	Solution
A printer produces gibberish or blank sheets.	The file contains data that cannot be printed.	Place the job on hold, drain and restart the printer, and notify the user who owns the job.
	A possible problem exists with a printer's toner or ribbon.	Check toner and ribbon and correct as needed.
Spooler response time is inexplicably slow.	System users report that their jobs are not being printed. Numerous unusual operator messages might occur.	Check with your management; a memory dump of the supervisor process might be needed.
The spooler does not accept jobs.	The collector queue file is full.	Delete or print jobs.
	The collector process is down.	Restart the collector process.
	The spooler supervisor process is down.	Check for error messages, stop any running collector or print processes, drain the spooler, and warmstart the spooler.
	Syntax was incorrectly entered.	Reenter syntax correctly.
	The spooler subsystem is hung.	Drain the spooler and perform a warmstart. If jobs still cannot be printed, notify your management; a memory dump of the supervisor process might be needed.

Table 14-7. Common Printer and Spooler Problems (page 2 of 2)

Problem	Possible Causes	Solution
Jobs fail to print.	A printer is down, offline, or in an error state.	Drain and restart the device in question, or manually put it online.
	The spooler subsystem is hung.	Drain the spooler and perform a warmstart.
	A print process is down or is in an error state.	Check configuration, restart the print process, and check device status.
	A paper jam has occurred.	Suspend the printer, physically fix the jam, and restart the printer.
	An invalid (Job -1) state has occurred with a given device.	Drain the device and restart it. If problems persist, drain and warmstart the spooler.
	A user sent a file to a dummy or nonexistent location (or put it on hold).	Reroute the job to an existing printer or take the job out of the hold state.
	FORM attributes of the job and the device do not match.	Reroute the job or change the FORM attributes.
	The spooler supervisor process is down.	Check for error messages, stop any running collector or print processes, drain the spooler, and warmstart the spooler.
	The job command fails to take effect (jobs hang).	Check with your management: a memory dump of the supervisor process might be needed.

15

Managing the Spooler Using Spoolcom

You manage the different components in spooler operations by adding, deleting, displaying, and modifying the attributes of collector processes, print processes, print devices, and locations:

These tasks are usually handled by operations management personnel.

You should be familiar with general Compaq *NonStop*™ Kernel system operations, spooler operations, and the spooler operations terms and concepts defined in Section 14, Performing Routine Spooler Operations Using Spoolcom.

Naming Spooler Components and Files

If you name a spooler component or file, observe the guidelines in Table 15-1, which allow for expansion of the file name across the network.

Note. All spooler components, including the supervisor, collectors, print processes, and data files for the collectors, should reside on a single system. Output devices can reside elsewhere.

Table 15-1. Spooler Naming Conventions

File or Component	Comments
Collector process name	The name is $*process* and has up to four alphanumeric characters. The first character is a letter. For example: $S2. There is no default name.
Collector program name	The default name is $SYSTEM.SYSTEM.CSPOOL. Replace the file if you want to run your own collector process.
Command files	The file name consists of eight alphanumeric characters. There is no default name.
Configuration file	The file name has up to eight alphanumeric characters. There is one file for each spooler, and no default name. For example: $SYSTEM.SPLUTIL.SPLCONF.
Control files	The file name has seven alphanumeric characters. The first character must be a letter. The spooler adds a numeric character to the end of the file name. There is no default name. For example: $MKT.SPL.SPL*n*.
Data files	The file name has eight alphanumeric characters. There is one file for each collector, and no default name. For example: $MKT.SPL.SPLDATA.
Device name	The name is $*device* and has up to six alphanumeric characters. The first character is a letter. There is no default name. For example: $SLOW.
Location name	The logical destination of a job, consisting of two names separated by a period—the group name and the destination name, as follows:
Group name	The group name is #*group* and has one through seven alphanumeric characters starting with a letter. If you do not specify a group name, your output is sent to the group named #DEFAULT. Group names must always begin with the # character.
Destination name	The destination name is *dest* and has up to eight alphanumeric characters. The first character is a letter. If you do not specify a destination name, your output is sent to the destination named DEFAULT.
Print process name	The name is $*process* and has up to four alphanumeric characters. The first character is a letter. There is no default name. For example, $USRP.
Print program name	The default name is $SYSTEM.SYSTEM.PSPOOL. Replace the file if you want to run your own print process.
Supervisor process name	The name is $*process* and has up to four alphanumeric characters. The first character is a letter. The default name is $SPLS.

Managing Collector Processes

Adding a Collector to Your Spooler Subsystem

You might want to add a collector if:

- The size of spooled jobs varies greatly
- Your applications sometimes get file-system error 45 (DISK DIRECTORY IS FULL)

Configuring several collectors with different unit sizes to handle the different types of jobs results in a more efficient use of disk space.

You can add a collector to a spooler any time the spooler is not draining or dormant. It is not necessary to stop the spooler to add a new collector. As soon as a new collector is defined, the supervisor control files are updated with its name and attributes. If you want the collector to be part of your configuration every time you system load the spooler, add it to the spooler configuration file.

1. Define the collector:

   ```
   > SPOOLCOM

   ) COLLECT $collector-name, DATA data-filename
             [ , collector-attribute ] ...
   ```

   Collector attribute subcommands are described in Table 15-2 on page 15-4.

2. If the spooler is in the warm state or cold state when you add the new collector, enter:

   ```
   ) SPOOLER, START
   ```

3. If the spooler is already active when you add the new collector, start the collector:

   ```
   ) COLLECT $collector-name, START
   ```

Example

This example is based on these collector attributes:

- The data file name is $SPOOL.SPOOLER.S2DATA.
- The backup CPU is processor 2.
- The processor that is to run this collector is CPU 3.

- The program file name is the default name ($SYSTEM.SYSTEM.CSPOOL).
- The execution priority is 144.
- The unit size is 8.

To add the collector $S2 with the above attributes to an active spooler subsystem, enter:

```
) COLLECT $S2, DATA $SPOOL.SPOOLER.S2DATA, 2, 3, 144, 8
) COLLECT $S2, START
```

Table 15-2. Collector Attributes

SPOOLCOM COLLECT Attributes & Subcommands	Description and Default Value
BACKUP *backup-cpu*	The processor that runs the collector backup. If you don't specify a BACKUP value, the collector runs without a backup.
CPU *cpu*	The processor that runs the collector. The default is for the collector to run on the same processor as the supervisor.
DATA *data-filename*	The name of the disk file where the collector stores jobs. You must specify this file name; there is no default name. The data file must be created before the collector is started. If you try to start a collector without a data file, the collector abnormally terminates with the error message: CANNOT OPEN SPOOLER DATA FILE
FILE *program-filename*	The program file for this collector process. If you don't specify a program file, the system runs a copy of $SYSTEM.SYSTEM.CSPOOL.
PRI *process-priority*	The execution priority of the collector (default is 145).
UNIT *unit-size*	The number of 512-word blocks requested by the collector when it needs more disk space (default is 4). You should set the unit size of a collector once and not change it. See the *Spooler Utilities Reference Manual*.

Displaying Collector Attributes

To display your collector's current attributes, enter:

```
) COLLECT $collector-name, STATUS DETAIL
```

Example

To display the current attributes of the collector named $S2:

```
) COLLECT $S2, STATUS DETAIL
```

```
COLLECTOR: $S2
    STATE: ACTIVE
    LAST ERROR: NONE
    PROGRAM FILE: $SYSTEM.SYSTEM.CSPOOL
    CPU: 3
    BACKUP: 2
    PRIORITY: 144
    DATA FILE: $SPOOL.SPOOLER.S2DATA
    PAGESIZE: 60
    UNIT SIZE: 8
    ALLOCATED UNITS: 1366
    TOTAL UNITS: 8192
    PERCENT FULL: 16
```

Modifying Collector Attributes

1. Drain the collector.

) COLLECT $*collector-name*, DRAIN

2. Modify the collector attributes.

) COLLECT $*collector-name*, *collector-attribute* ...

   Collector attributes and COLLECT subcommands are described in Table 15-2 on page 15-4. If you do not specify attributes, they take on a default value.

   You should not change the unit size of an existing collector. If a different unit size is required, delete the old collector and start a new one.

   Do not change the data file where the collector stores jobs; jobs can be lost.

3. Restart the modified collector:

) COLLECT $*collector-name*, START

Example

To modify a collector attribute by changing the backup processor from CPU 2 to CPU 5:

```
> SPOOLCOM
) COLLECT $S2, DRAIN
) COLLECT $S2, BACKUP 5
) COLLECT $S2, START
```

Deleting a Collector

1. Drain the collector:

) COLLECT $*collector-name*, DRAIN

2. Delete the collector from the spooler subsystem:

```
) COLLECT $collector-name, DELETE
```

Example

To delete collector $S2 from the spooler:

```
> SPOOLCOM
) COLLECT $S2, DRAIN
) COLLECT $S2, DELETE
```

Managing Print Processes

You can use a print process provided by Compaq, or write your own. Compaq-provided print processes all reside on $SYSTEM.SYSTEM. If you write your own print process, see the *Spooler Programmer's Guide*.

Table 15-3. Compaq-Provided Print Processes

Process	Description
FASTP	Combines PSPOOL and PSPOOLB features and provides performance improvements for many types of printers. Each FASTP print process supports a maximum of 32 active devices. This process is recommended.
PSPOOL	Sends jobs to a print device, one line at a time. A PSPOOL can work with all devices. Each PSPOOL print process can support up to 32 active devices.
PSPOOLB	Supported the SNAX CRT and PRT interfaces before FASTP was released. PSPOOLB is primarily used with a SNAX line that includes one or more 6600 controllers. A PSPOOLB stores jobs in an internal buffer. When the buffer is full, it sends these jobs, one line at a time, to the SNAX interface; so PSPOOLBs are faster than PSPOOLs. Each PSPOOLB can support up to 12 active devices, but you should limit your number of devices to 8.

Adding a Print Process

Adding a print process to a spooler can reduce the response time of the spooler and the printers associated with it. You can have 1 through 45 print processes within each of your spooler supervisor systems. You can add a print process to a spooler any time the spooler is not draining or dormant. To learn the number of print processes currently configured for your spooler, enter the SPOOLCOM PRINT, STATUS command.

To add a print process to an existing spooler, then associate a device with it:

1. Specify the print process parameters:

```
> SPOOLCOM
) PRINT $process-name, FILE $SYSTEM.SYSTEM.name, CPU n
```

2. Associate that print process with a device:

```
) DEV $device, PROCESS $process-name
```

If you want a print process to be part of your configuration every time you coldstart the spooler, be sure to add it to your spooler configuration file.

See the *Spooler Utilities Reference Manual* for complete descriptions of the SPOOLCOM PRINT and DEV commands.

Examples

- To define a new print process named $XP to be a copy of the program file $SYSTEM.SYSTEM.PSPOOL and to run on CPU 3; the process then readies the new print process to run jobs:

```
> SPOOLCOM
) PRINT $XP, FILE $SYSTEM.SYSTEM.PSPOOL, CPU 3
```

- To add a PSPOOLB print process called $PPPS to an existing spooler on processor 2:

```
) PRINT $PPPS, FILE $SYSTEM.SYSTEM.PSPOOLB, CPU 2
```

- To add a FASTP print process called $XLNT to an existing spooler on processor 4:

```
) PRINT $XLNT, FILE $SYSTEM.SYSTEM.FASTP, CPU 4
```

Displaying the Current Attributes of a Print Process

To display the current attributes of a print process, enter:

```
) PRINT $print-process, STATUS DETAIL
```

Example

To display the current attributes of the print process named $SPLA, enter:

```
) PRINT $SPLA, STATUS DETAIL
```

A listing such as this is displayed on your home terminal:

```
PRINT PROCESS: $SPLA
    STATE: ACTIVE
    LAST ERROR: NONE
    DEBUG: OFF
    INDEPENDENT: NO
    PROGRAM FILE: $SYSTEM.SYSTEM.PSPOOL
    CPU: 3
    BACKUP: 2
    PRIORITY: 128
    PARM: 0
```

Modifying Print Process Attributes

You can modify the attributes of a print process when it is in a dormant or procerror state:

1. Find out which devices the print process controls:

   ```
   ) PRINT $print-process, XREF
   ```

2. Drain the device:

   ```
   ) DEV $device-name, DRAIN
   ```

3. Start the print process with the new attributes:

   ```
   ) PRINT $print-process [ , print-process-attribute ]
   ```

   Print process attributes are described in Table 15-4 on page 15-10.

Example

To modify an attribute of the print process named $SPLA by changing the backup processor from 2 to 4:

1. Enter:

   ```
   ) PRINT $SPLA, XREF
   ```

   A listing such as this is displayed on your home terminal:

```
PRINT PROCESS       DEVICE              LOCATION
$SPLA               $LP1                #FAST.DEFAULT
$SPLA               $LP2                #SLOW.DEFAULT
```

2. Check to see if all devices are offline (not busy or waiting):

   ```
   ) DEV $LP1; DEV $LP2
   ```

   A listing such as this is displayed on your home terminal:

```
DEVICE              STATE       FLAGS  PROC     FORM
$LP1                WAITING            $SPLA

DEVICE              STATE       FLAGS  PROC     FORM
$LP2                OFFLINE            $SPLA
```

3. Drain the waiting device $LP1:

   ```
   ) DEV $LP1, DRAIN
   ```

4. Check the status of $LP1 to confirm it is offline:

   ```
   ) DEV $LP1
   ```

   A listing such as this is displayed on your home terminal:

```
DEVICE              STATE       FLAGS  PROC     FORM
$LP1                OFFLINE            $SPLA
```

5. Check the status of print process $SPLA:

   ```
   ) PRINT $SPLA
   ```

   A listing such as this is displayed on your home terminal:

```
PRINT    STATE            FLAGS   CPU    PRI
$SPLA    DORMANT                  3 , 2  128
```

6. Change the backup processor:

   ```
   ) PRINT $SPLA, BACKUP 4
   ```

7. Start the device:

   ```
   ) DEV $LP1, START
   ```

8. Check the status of the print process:

   ```
   ) PRINT $SPLA
   ```

   A listing such as this is displayed on your home terminal:

```
PRINT    STATE            FLAGS   CPU    PRI
$SPLA    ACTIVE                   3 , 4  128
```

Deleting a Print Process From the Spooler

1. Find out which devices the print process controls:

   ```
   ) PRINT $print-process, XREF
   ```

2. For each device, set DEV $*device* EXCLUSIVE OFF so that the print process will close the device when no more jobs are waiting to be printed:

   ```
   ) DEV $device-name, EXCLUSIVE OFF
   ```

3. For each device, drain the device:

   ```
   ) DEV $device-name, DRAIN
   ```

4. Delete the print process:

   ```
   ) PRINT $print-process, DELETE
   ```

   The print process should become dormant approximately two minutes after the process prints its last job. If the print process does not become dormant at this time, issue a TACL STOP command to stop the process.

   Be sure to connect another print process to the devices before you start them again.

Example

To check the locations associated with print process $SPLB, drain the device connected to it, and delete $SPLB from the spooler:

1. Find out which devices print process $SPLB controls:

   ```
   ) PRINT $SPLB, XREF
   ```

   A listing such as this is displayed on your home terminal:

```
PRINT PROCESS       DEVICE              LOCATION
$SPLB               $LP1                #FAST.DEFAULT
```

2. Drain the device $LP1 connected to the print process:

   ```
   ) DEV $LP1, DRAIN
   ) DEV $LP1
   ```

   A listing such as this is displayed on your home terminal:

```
DEVICE              STATE       FLAGS  PROC     FORM
$LP1                OFFLINE             $SPLB
```

3. Delete print process $SPLB:

   ```
   ) PRINT $SPLB, DELETE
   ```

4. Connect device $LP1 to a different print process:

   ```
   ) DEV $LP1, PROCESS $SPLC
   ```

5. Restart device $LP1:

   ```
   ) DEV $LP1, START
   ```

Print Process Attributes

You define the attributes of a print process with the SPOOLCOM PRINT command. Any print process attributes not specified in the PRINT command take a default value.

Table 15-4. Print Process Attributes and PRINT Subcommands (page 1 of 2)

SPOOLCOM PRINT Attributes and Subcommands	Description and Default Value
BACKUP *backup-cpu*	The processor that runs the print process backup. This attribute is used for user-written, independent print processes only. An independent print process is one that is running when the spooler is started. The spooler does not start it and assumes that it is always running. Print processes supplied by Compaq do not run as process pairs, so the BACKUP subcommand is ignored.
CPU *cpu*	The processor that runs the print process; by default the same processor as the supervisor.
DEBUG [OFF]	Sets the Debug mode of the print process (default is Debug OFF). Print processes should never run in Debug mode on a production spooler.

Table 15-4. Print Process Attributes and PRINT Subcommands (page 2 of 2)

SPOOLCOM PRINT Attributes and Subcommands	Description and Default Value
FILE *program-filename*	The program file for this print process (by default assumes this is an independent process).
PARM *parameter*	A print process parameter passed by the supervisor to the print process in the startup message. Its meaning is defined by a user-written print process, and it can be used only with user-written print processes.
PRI *process-priority*	The execution priority of the print process (default is 145).

Managing Print Devices

If a device has been configured but was not included in the spooler coldstart, you can add it to your spooler any time the spooler is not draining or dormant. It is not necessary to stop the spooler to add a new device. As soon as a device is defined, the supervisor control files are updated with the name and corresponding attributes of the new device.

Note. The total number of devices must not exceed the maximum configured at the time of spooler coldstart, and the number of locations cannot exceed that maximum.

If you want a device to be part of your configuration every time you perform a system load of the spooler, be sure to add it to the spooler configuration file.

Adding a Print Device To Your Spooler Subsystem

1. Enter this SPOOLCOM command:

```
> SPOOLCOM
) DEV $device, PROCESS $print-process
      [ , device-attribute ] ...
```

   $*device* can be a virtual device, a process, a file, or the name of a device that is configured onto your system. (Use SCF LISTDEV to find out the names of all the devices configured onto your system.)

   $*print-process* is the name of the print process that controls the device.

Device attribute subcommands are described in Table 15-5 on page 15-15.

2. Establish the connection between the location and the device:

   ```
   ) LOC [ #group. ] dest, DEV $device
   ```

3. Bring the device online:

   ```
   ) DEV $device, START
   ```

   The total number of devices including this one must not exceed the maximum configured at the time of spooler coldstart, and the number of locations cannot exceed that maximum.

Example

To add a device named $LP3 to the spooler and cause all jobs printed on that device to have a standard header page, enter:

```
> SPOOLCOM
) DEV $LP3, PROCESS $XP, HEADER ON
) LOC #LINE.DEFAULT, DEV $LP3
) DEV $LP3, START
```

Displaying Current Print Device Attributes

To display the current attributes of a print device, enter:

```
) DEV $device-name, STATUS DETAIL
```

Example

To display the current attributes of the device named $LP, enter:

```
) DEV $LP, STATUS DETAIL
```

A listing such as this is displayed on your home terminal:

```
DEVICE: $LP
    STATE: BUSY
    LAST ERROR: %004014
    EXCLUSIVE: OFF !
    FIFO: OFF
    HEADER: ON
    TRUNCATION: OFF
    DRAINING: NO
    PRINTING JOB: 1199
    PARM: 0
    PROCESS: $SPLA
    RETRY: 5
    TIMEOUT: 4096
    SPEED: 700
    WIDTH: 900
    FORM:
    RESTART: 120
    DEVRESET: OFF
    DEVTYPE:
    STARTFF: OFF !
    ENDFF: ON
    CHARMAP: NONE
    PREXLATE: OFF
    LUTOFVALUE: CRFFCR
    LUEOLVALUE: CRLF
    LUEOLUHEN: LT132
```

Modifying Print Device Attributes

1. Drain the device to take it offline:

) DEV $*device*, DRAIN

   Device attributes can be changed only when the device is offline.

2. Specify the device attributes:

) DEV $*device device-attribute ...*

   Device attributes are described in Table 15-5 on page 15-15.

3. Put the device back online:

) DEV $*device*, START

Example

To change the form name assigned to the device named $PRINT, enter:

```
) DEV $PRINT, DRAIN
) DEV $PRINT, FORM LETTERH
) DEV $PRINT, START
```

Now, only jobs that specify form LETTERH will print on this device; all other jobs sent to this device remain in the device queue.

Deleting a Print Device

1. Drain the device. Draining the device allows the job currently printing to finish printing but stops new jobs from starting on the device.
2. Empty the device queue:
   a. Break the connection between the device and its locations.
   b. Delete the locations (if there are no jobs waiting).
   c. Reroute the locations to a different device (if there are jobs waiting to print).

Deleting a Device From a Running Spooler

1. Enter the SPOOLCOM DEV, DRAIN command to drain the device:

```
> SPOOLCOM
) DEV $device, DRAIN
```

   This causes the device to go offline after the job currently printing is finished.

2. Empty the device queue by doing one of:
   a. Break the existing connection between the device and its location:

```
) LOC #group.dest, DEV
```

   or

   a. See if there are any jobs at the location:

```
) LOC #group.dest, STATUS
```

   b. If there are no jobs currently in the location, delete the location:

```
) LOC #group.dest, DELETE
```

   c. If there are jobs waiting to print, reroute the location to a different device so those jobs can finish printing:

```
) LOC #group.dest, DEV $different-device
```

3. Delete the device from the spooler system:

```
) DEV $device, DELETE
```

Example

To delete the printer named $LP1 from a running spooler by first draining it and then disconnecting it from its location, enter:

```
) DEV $LP1, DRAIN
) LOC #LP1.LP1, DEV
) DEV $LP1, DELETE
```

Print Device Attributes

You define the attributes of a print device with the SPOOLCOM DEV command. Any device attributes not specified in the DEV command take on a default value.

Table 15-5. Print Device Attributes and DEV Subcommands (page 1 of 3)

SPOOLCOM DEV Attributes and Subcommands	Description and Default Value
DEVRESET [ON | OFF]	DEVRESET and DEVRESET ON reset IOP print processes back to their values configured during SYSGEN without having to reset the printer. The default value, DEVRESET OFF, causes the print process to reset the print IOP, but not the print device, to its configured default values at the beginning of each print job.
DEVTYPE [*blank* | LU1 | LU3 | 7 | 8 | 9 | 10 | 5512 | 5515 | 5516 | 5573 | 5574 | 5577]	Controls the configuration of a print device. Applies to the FASTP and PSPOOLB print processes. For an explanation of DEVTYPE options, see the *Spooler Utilities Reference Manual*.
ENDFF [ON | OFF]	ENDFF or ENDFF ON causes PSPOOL or FASTP to issue a form feed at the end of a job. ENDFF OFF, the default value, causes PSPOOL or FASTP to suppress a form feed at the end of a job.
EXCLUSIVE [ON | OFF [!]]	Specifies the ownership mode of the device with respect to the spooler: • EXCLUSIVE or EXCLUSIVE ON specifies that the print process should not close the device when the printer is not printing a job. The print process keeps the device open all the time, preventing any other process from gaining access. • EXCLUSIVE OFF (the default) specifies that the print process should close the device when no more jobs are waiting to be printed. This lets other processes access the device. • EXCLUSIVE OFF ! specifies that the device is to be closed between jobs.
FIFO [ON | OFF]	Specifies the algorithm by which jobs are selected for printing: FIFO or FIFO ON indicates a first-come, first-served basis. FIFO OFF (the default) causes jobs to be added to the device queue according to an algorithm that takes into account the relative lengths of jobs already in the queue.
FORM [*form-name*]	Specifies the form name for the device. Specifying a form name guarantees that only certain types of jobs print on the device. This attribute is most commonly used when the device is loaded with special paper or ribbon. The default name is a blank form name.

Table 15-5. Print Device Attributes and DEV Subcommands (page 2 of 3)

SPOOLCOM DEV Attributes and Subcommands	Description and Default Value
HEADER [ON | OFF | BATCH]	Specifies whether a standard or batch header page appears at the beginning of every job. HEADER or HEADER ON (the default) specifies that a standard header page prints at the beginning of every job. The standard header contains the job report name, location, and job number. HEADER BATCH specifies that two header pages and three trailer pages print for every job. The PSPOOL and FASTP print processes support the BATCH option; PSPOOLB does not support BATCH.
LUEOLVALUE [NL | CRLF]	Sets the end-of-line (EOL) sequence that the FASTP print process places at the end of a print record for LU1 and LU3 type printers only.
LUEOLWHEN [LT132 | LTWIDTH | ALWAYS | NEVER]	Indicates when the FASTP print process is to place an end-of-line (EOL) sequence at the end of a print record for LU1 and LU3 type printers.
LUTOFVALUE [CRFFCR | FFCR | FF | NEVER]	Sets the sequence that designates top of form (new page) that the FASTP print process is to use for a print record for LU1 and LU3 type printers.
PARM *parameter*	Specifies a device parameter passed to the print process controlling this device whenever the supervisor communicates with the print process. The range is -32768 through +32767. The default is 0, which specifies default values for parameters. PARM applies to a PSPOOLB or FASTP print process; it has no meaning for PSPOOL. For details, see the *Spooler Utilities Reference Manual.*
PROCESS $*process-name*	Specifies the print process that controls the device. You must specify a print process; there is no default process.
RESTART [OFF | *interval* | ON]	RESTART or RESTART ON specifies the number of seconds the device waits between automatic restart attempts. The default value is OFF, meaning that device restarts are not attempted.
RETRY *interval*	Specifies the number of seconds the print process waits before retrying a failed retryable write to the device. The default value is 5 seconds.
SPEED *lines-per-min*	A value used by the supervisor to calculate how long jobs take to print on the device. This attribute is used only for queuing jobs; it has no effect on the device printing speed. The default is 100 lines per minute.
STARTFF [ON | OFF [!]]	STARTFF or STARTFF ON causes the PSPOOL and FASTP print processes to issue a form feed at the beginning of a job. STARTFF OFF, the default value, and STARTFF OFF ! cause these print processes to suppress a form feed at the start of each job. STARTFF has no meaning for PSPOOLB.

Table 15-5. Print Device Attributes and DEV Subcommands (page 3 of 3)

SPOOLCOM DEV Attributes and Subcommands	Description and Default Value
TIMEOUT *number-of-retries*	Specifies the number of times the spooler retries a failed write to the device. The default value is 360 times.
TRUNC [ON | OFF]	Tells the standard print process whether to truncate or wrap around lines longer than the device width. The default value is TRUNC OFF, which means that the ends of long lines are printed on the next sequential line.
WIDTH *device-width*	Specifies the maximum line length for the device. If you do not specify a line length, the print process obtains the record size with a DEVICEINFO procedure call. (See what this is by using SCF STATUS for the device and looking at the record size column.) The range of values is 1 through 32767: the default value is -1.

Managing Locations

You can add a location to a spooler, and display and modify location attributes from your spooler subsystem at any time. To delete a location, there must not be any jobs currently in that location.

Adding a Location and Connecting It to a Device

To add a location to your spooler subsystem and connect it to a device, enter:

```
> SPOOLCOM
) LOC [ #group. ] dest , DEV $device
```

Example

To connect location #PRIN.DEFAULT to device $LP and location #PRIN.CHAR to device $PRINT, enter:

```
) LOC #PRIN.DEFAULT, DEV $LP
) LOC #PRIN.CHAR, DEV $PRINT
```

Displaying a Location's Current Attributes

To display the current attributes of a given location, enter:

```
) LOC #loc, STATUS DETAIL
```

For more information about location attributes, see Table 15-6 on page 15-18.

Example

To display the current attributes of the location named #BIRD6.DEFAULT, enter:

```
) LOC #BIRD6.DEFAULT, STATUS DETAIL
```

A listing such as this is displayed on your home terminal:

```
LOCATION: #BIRD6.DEFAULT
    BROADCAST: OFF
    DEVICE: \CAT.$C
     FONT NAME:
```

This listing shows that the BROADCAST attribute is off, that this location is connected to the device \CAT.$C, and that no FONT attribute has been configured for this location.

Table 15-6. Location Attributes

SPOOLCOM LOC Attributes and Subcommands	Description and Default Value
BROADCAST [ON | OFF]	BROADCAST or BROADCAST ON causes a job routed to #*group* to be printed on all devices connected to #*group*. BROADCAST OFF, the default value, causes jobs routed to #*group* to be printed on the one device connected to the group that can print the job the fastest.
DEV $*device*	Specifies which device the location is connected to. There is no default; you must connect a location to a device to direct the jobs sent to the location to an output device.
FONT *font-name*	Specifies which font job should be downloaded to the location before any jobs are sent to it. This attribute causes jobs routed to a particular location to be printed with the character set, compressed print mode, or whatever other printing attributes have been programmatically set in the *font-name* file. See the *Spooler Utilities Reference Manual* for details and instructions.

Modifying Location Attributes

To modify location attributes by breaking the connection between a location and a device, enter:

```
) LOC [ #group.] dest, DEV
```

If you specify only the destination, the command refers to every group that has that destination in it.

Example

To modify location attributes by breaking the connection between the location #BIRD6 and any devices connected to that location, enter:

```
) LOC #BIRD6, DEV
```

Deleting a Location

1. Check that there are no jobs currently in the location being deleted:

   ```
   ) LOC #group [.dest ]
   ```

   The status of any jobs currently in the location are displayed. If any jobs are listed, you must delete them or wait until they finish before you can delete the location.

2. Delete the location from the spooler:

   ```
   ) LOC #group [.dest ] , DELETE
   ```

   If you specify only the group, this command refers to all locations within the group.

Example

To check the status of current jobs, then delete the location #BIRD6, enter:

```
) LOC #BIRD6
```

A listing such as this is displayed on your home terminal:

```
LOCATION          FLAGS  DEVICE                              FONT
#BIRD6.DEFAULT             \CAT.$C
```

This listing shows no current jobs in this location, so you can delete it:

```
) LOC #BIRD6, DELETE
```

Rebuilding the Spooler Control Files

A spooler is characterized by its control files, which contain the names and attributes of the processes, locations, and devices that make up the system.

If a spooler was improperly drained or suffered an abnormal shutdown, you must rebuild the spooler control files:

1. Warmstart the spooler (see Warmstarting a Drained Spooler on page 14-12).
2. Enter the SPOOLCOM JOB command:

   ```
   ) JOB
   ```

   If you find a Job 0 listed as a result of the SPOOLCOM JOB command, the spooler has a corrupted job map.

3. Rebuild the spooler control files and the job bit map by executing the same SPOOL command you normally use for a warmstart but add the REBUILD option after the supervisor process name, and specify new values. These values should be the same as those specified in your coldstart command file for the maximum number of:

   - Jobs that the spooler handles at any one time (*num-of-jobs*)
   - Destinations and groups the routing structure contains (*num-of-locations*)

- Devices that can be known to the spooler at any one time (*num-of-devices*)
- Collectors that can be declared for this spooler (*num-of-collectors*)
- Print processes that can be declared for this spooler (*num-of-print-processes*)
- Font jobs that can be declared for this spooler (*num-of-font-jobs*)
- Batch jobs that can be declared for this spooler (*num-of-batch-jobs*)

a. Drain the spooler:

```
> SPOOLCOM SPOOLER, DRAIN
```

b. Rebuild the spooler control files:

```
> SPOOL / IN control-filename ,
          NAME $supervisor-process-name /
            R[EBUILD]
            , num-of-jobs , num-of-locations
            , num-of-devices , num-of-collectors
            , num-of-print-processes
            , num-of-font-jobs
            , num-of-batch-jobs
```

You must include the REBUILD option if you want to increase the values of any coldstart parameters (*num-of-jobs* and so forth) in a warmstart. If you do not use this option, you must coldstart the spooler to increase the values of any parameters. You cannot decrease the values of any parameters. See the *Spooler Utilities Reference Manual* for more information about SPOOL.

c. Add, delete, or modify spooler components as needed.

4. Enter SPOOLCOM and start the spooler:

```
> SPOOLCOM SPOOLER, START
```

Example: Rebuilding a Warmstarted Spooler's Control Files

This example uses the spooler in Warmstarting a Drained Spooler on page 14-12, and:

- $MKT.SPL.SPL is the control file.
- $SPLS is the name of the supervisor.
- R stands for REBUILD.
- 995 is the maximum number of jobs and locations for this spooler.
- 150 is the maximum number of devices for this spooler.
- 10 is the number of collectors and print processes for this spooler.

```
> SPOOLCOM SPOOLER, DRAIN
> SPOOL /IN $MKT.SPL.SPL, NAME $SPLS/ R,995,995,150,10,10
> SPOOLCOM SPOOLER, START
```

Part V. Security Features and Other Guardian Utilities

This part of the guide contains information about the security features available through FUP and TACL, and instructions for using the Guardian utilities DSAP, Error, and VPROC:

- Section 16, Managing Users and Security
- Section 17, Monitoring Event Messages
- Section 18, Displaying Version and System Information

16 Managing Users and Security

Ensuring that the status of all system users is current is an important daily task to maximize Compaq *NonStop*™ Kernel system usage. This includes making sure access is set for all users as needed, and helping to solve any problems that may arise for them. The responsibility for supporting users is sometimes handled by system administrators, but these responsibilities are sometimes also handled by operators, depending on the work environment.

This section describes routine tasks to assist users of the Guardian environment. It also describes the security features of the Guardian environment as they apply to general (or application) users – users who log on to a local or remote system to run an application program such as electronic mail or a text editor.

This section contains the following information for supporting your NonStop™ Kernel system users:

Note. Depending upon your company's policies and system configuration, you might not be able to perform all of the tasks included in this section, especially if your system is protected by Safeguard. Consult with your security administrator or system manager if you have questions about your system or network. The *Security Management Guide* also contains helpful information about tasks and issues involving system security.

Your Responsibility to System Users

When overseeing Guardian operations on a NonStop™ Kernel system, you have certain responsibilities to the system users to ensure they have as few problems as possible:

Keeping Current

Depending on your company's policies, when you arrive at work you might be expected to check for:

- Messages from the operations staff who worked the shift before yours. If you have a shift log book, be sure to check for any messages left by these operators.
- Telephone messages, electronic mail, faxes, and so on from users of your system. Respond to these messages promptly: system users are often the first to notice a potential problem.
- Operator messages that have occurred during the shift before yours and respond to any potential problems (see Section 17, Monitoring Event Messages).

Section 1, Introduction to Guardian System Operations, contains a complete check list of tasks operators should perform at the beginning of each shift. The above list represents areas that are of particular importance when supporting system users.

Monitoring the System Frequently

All system operations tasks help the users of your system in some way. With this in mind, the importance of monitoring your system's status on a regular basis should become more clear. Section 19, Monitoring Hardware Components, includes specific instructions for checking the status of various system components.

You should monitor:

- This hardware: disks, processors, terminals, tape drives, printers
- The status of communication lines that connect hardware components to the system
- Key applications
- Key subsystems, for example the spooler
- System processes

Adding Users to the System

To add new users to your system, use the TACL ADDUSER program:

Note. If your system is protected by the Safeguard security product, the ADDUSER program might not be available for system operators to use.

---

1. Log on as the super ID (255,255) or as a group manager.

If you are authorized as the super ID (255,255), you can add a new user to any group on your local system. You can also add a new group name at the same time you add a new user. If you are a group manager (*n*,255), you can add a new user to your own group.

2. Enter the ADDUSER command at your TACL prompt:

   ```
   > ADDUSER group-name.user-name, group-id, user-id
   ```

   where *group-name* and *user-name* are the group and individual names, respectively, of the new user. Each name can contain from one through eight letters or digits, and the first character must be a letter.

   group-id is an integer in the range 0–255 that uniquely identifies a group. Note that 255 is reserved as the *group-id* for system operators and for the super ID.

   user-id is an integer in the range 0–255 that uniquely identifies a user within a group. 255 is reserved as the *user-id* for group managers and the super ID.

3. Assign a password and default volume and subvolume for the new user:

   ```
   > LOGON group-name.user-name
   Password: (RETURN)
   > PASSWORD password
   > DEFAULT $volume.subvolume, "security-code"
   ```

   If you do not specify any in this step, the logon defaults for users when they are first added to the system are:

   - Default volume: $SYSTEM
   - Default subvolume: NOSUBVOL
   - Default disk file security: AAAA
   - No password is assigned.

You can include options of the TACL RUN command in the ADDUSER program syntax. See the RUN[D] command description in the *TACL Reference Manual* for details about RUN options.

Examples

- To create a new group at the same time you add a new user, enter:

  ```
  > ADDUSER SALES.BONNIE, 8,1
  ```

  ```
  SALES.BONNIE (8,1) HAS BEEN ADDED TO THE USERID FILE.
  ```

  This command creates a new group named SALES, with the group ID 8, and also adds a new user named BONNIE, with user ID 1.

- The Super ID can also add a manager to the SALES group by adding a user with the user ID 255. For example, to add the manager BOSS, enter:

```
> ADDUSER SALES.BOSS, 8,255
```

```
SALES.BOSS (8,255) HAS BEEN ADDED TO THE USERID FILE.
```

- The new manager, BOSS, can add other members to the SALES group. For example:

```
> ADDUSER SALES.PAT, 8,2
```

```
SALES.PAT (8,2) HAS BEEN ADDED TO THE USERID FILE.
```

```
> ADDUSER SALES.DANA, 8,3
```

```
SALES.DANA (8,3) HAS BEEN ADDED TO THE USERID FILE.
```

Changing Logon Defaults

If the user SALES.BONNIE wants to change her default volume, subvolume, and file security, she would enter:

```
> LOGON SALES.BONNIE
Password: <password>
> DEFAULT $DATA1.BONNIEF, "NUUU"
> LOGOFF
```

The default volume and subvolume for SALES.BONNIE are now $DATA1.BONNIEF. The security for all disk files that SALES.BONNIE creates in the new default volume and subvolume will be NUNU.

For more information, see the *TACL Reference Manual* about the DEFAULT program and the *File Utility Program (FUP) Reference Manual* about file security settings.

About Passwords

Rules regarding the correct number of characters required for passwords can vary from system to system. Check with your operations manager if you have questions about rules regarding passwords that might be specific to your system.

A password must not include blanks, commas, or null characters. Passwords are case-sensitive; that is, those letters that are in uppercase the first time you enter a new password must appear in uppercase whenever you enter that password (and the same is true for lowercase letters).

For systems protected by the Safeguard security product, considerations and rules regarding passwords can vary. For more information, see the *Safeguard User's Guide.*

If users report that they have forgotten their passwords, your operations manager or security administrator can help restore these users' access to the system.

Deleting Users From the System

When you delete users from a system, either because they start to work on a different system or because they have left the company, you should first delete their accounts and then clean up their files and subvolumes:

Task 1: Delete the User Account

Use the TACL DELUSER program to delete the user's account from the system. The Super ID (255, 255) can delete any user from any group in the system, but group managers (*n*,255) can add users to and delete users from only their own groups.

Note. If your system is protected by the Safeguard security product, the DELUSER program might not be available for system operators to use.

---

1. Log on as the Super ID (255, 255) or a group manager (*n*,255).
2. At your TACL prompt, enter:

   ```
   > DELUSER group-name.user-name
   ```

   where *group-name* and *user-name* are the group and individual names, respectively, of the user who is to be deleted. Each name can contain from one to eight letters or digits, and the first character must be a letter.

You can include options of the TACL RUN command in the DELUSER program syntax. See the *TACL Reference Manual* for details on using the RUN command.

Example

In this example, the group manager of the SALES group or a super-group user (255,*n*) can delete the user SALES.BONNIE and other members of that group.

To delete the user SALES.BONNIE, enter:

```
> DELUSER SALES.BONNIE
```

This message is displayed:

```
SALES.BONNIE (8,1) HAS BEEN DELETED FROM THE USERID FILE.
```

Task 2: Clean Up the User's Disk Space

Because old or unneeded files use disk space and can constitute a security problem, you should delete or move the files and subvolumes owned by a deleted user. Encourage users to purge or move their subvolumes and files before you delete them from the system. If they cannot perform this task, their managers might ask you to help clean up the subvolumes and files:

1. Run a Disk Space Analysis Program (DSAP) report on each user's subvolumes on a specified disk volume:

   ```
   > DSAP $disk, BYSUBVOL USER group-number,user-number
   ```

DSAP returns a report that displays all subvolume names owned by the specified user, a listing of files, total pages used, and so forth.

2. Determine the subvolumes you want to purge to free some space on the specified disk. If necessary, consult with the deleted user's manager.

   You can print the DSAP report to review:

   ```
   > DSAP / OUT $disk.#printer / $disk, BYSUBVOL USER
   group-number,user-number
   ```

3. Delete the files determined in Step 2. If you are not logged on as the Super ID (255,255), you might not have authority to delete these files. In that case, ask the user's group manager (or the Super ID) to delete the files.

For additional instructions and examples of using DSAP, see Managing Disk Space Usage on page 9-14. Complete syntax, considerations, and examples for the DSAP program are in the *Guardian Disk and Tape Utilities Reference Manual*. For more information about FUP commands and considerations relating to their use, see the *File Utility Program (FUP) Reference Manual*.

Example

1. Run a DSAP report on the subvolumes and files owned by SALES.BONNIE (user 8,1) on \SAGE.$DATA1:

   ```
   > DSAP $DATA1, BYSUBVOL USER 8,1
   ```

   A report similar to this is returned to your home terminal:

```
Disk Space Analysis Program -- T9543D20 - T9543D20 - (01JUN93)  -- 7/20/93
10:27:22
Tandem Computers Incorporated 1981, 1983, 1985-1993

Summary of space use for SALES.BONNIE on $DATA1

         234 allocated pages in 8 files in 19 extents (0.0%).

          24 unused pages in 7 files (0.0%).

           0 deallocatable extent pages in 0 files (0.0%).

No SQL views.

Subvol Summary Report

 Subvolume Name                 Total Unused Dealloc Large  Min Age  Num
                         Files  Pages  Pages  Pages   File  Mod,Opn  Exp
BONNIEF                     6    162    23      0     140    6,  0    6
MEMOS                       2     72     1      0      64    0,  0    2
```

   This report shows that SALES.BONNIE has eight files in two subvolumes.

2. Consult with the manager of SALES.BONNIE before you purge any files or subvolumes. To show this manager a printout of the DSAP report, send the report to the printer $S.#LASER1:

   ```
   > DSAP / OUT $S.#LASER1 / $DATA1, BYSUBVOL USER 8,1
   ```

3. Delete the files.

Determining Group and User Name and Number

You sometimes need to learn users' names to notify them of problems with programs they control. Use the TACL USERS program on your local system to learn a user's group and user names if you only have the group and user numbers, and vice versa.

Note. If your system is protected by the Safeguard security product, the USERS program might not be available for system operators to use.

If you know a user's group and user ID numbers...

To learn the user's name, group name, and default volume and subvolume, enter:

```
> USERS group-id,user-id
```

If you know a user's group ID number only...

To learn the user's name, group name, user ID number, and default volume and subvolume, enter:

```
> USERS group-id, *
```

If you know a user's group name only...

To learn the user's name, ID number, and default volume and subvolume, enter:

```
> USERS group-name.*
```

Examples: Getting User Information

Getting User Name and Information for a User ID

To learn the user name and other information associated with the user ID 8,1, enter:

```
> USERS 8,1
```

USERS displays information such as:

```
GROUP       USER        I.D. #        SECURITY         DEFAULT VOLUMEID
SALES      .BONNIE     008,001          NUNU            $DATA1.BONNIEF
```

Getting User ID and Information for a User Name

To learn the user ID and other information about the user SALES.BONNIE, enter:

```
> USERS SALES.BONNIE
```

USERS displays the same information shown in the previous example.

Getting Information For All Users in a Group

To get information for all users in the SALES group (group ID 8), enter either:

```
> USERS SALES.*
```

or

```
> USERS 8, *
```

USERS displays information about all users in the SALES group, such as:

```
GROUP       USER        I.D. #        SECURITY         DEFAULT VOLUMEID

SALES      .BONNIE     008,001          NUNU            $DATA1.BONNIEF
SALES      .PAT        008,002          UUUU            $DATA1.PATANDR
SALES      .DANA       008,003          NUUO            $DATA1.DANAC
SALES      .BOSS       008,255          OOOO            $DATA1.HONCHO
```

See the *TACL Reference Manual* for more information about the USERS program.

Interfaces for the Security Features

Compaq provides several user interfaces for the Guardian security features:

- The TACL program

  TACL commands and programs provide user and logon security.

- File Utility Program (FUP)

  FUP commands provide security for disk files.

- Peripheral Utility Program (PUP) (D-series only)

  PUP commands provide security for disks and other peripheral devices. For example, the PUP ALLOWOPENS command permits users to open files on a disk volume. PUP is primarily used by system operators (user ID 255,*n*). For more information, see the *Peripheral Utility Program (PUP) Reference Manual*.

- Safeguard subsystem

  Safeguard software provides additional security features for systems and distributed networks. Instructions for logging on at a terminal controlled by Safeguard software are included in Section 2, Getting Started With TACL. For more information about the Safeguard subsystem, see the *Safeguard User's Guide*.

Table 16-1. TACL System Security Features (page 1 of 2)

Command or Program	Function
ADDUSER program	Adds new users to the system (user ID must be *n*,255)
DEFAULT program	Sets system, volume, subvolume, and disk-file default security attributes (RWEP)
DELUSER program	Deletes users from the system (user ID must be *n*,255)
LOGOFF command	Terminates communication with a TACL process
LOGON command	Establishes communication with a TACL process
PASSWORD program	Selects, changes, or deletes a local password

For a detailed description of these commands and programs, see the *TACL Reference Manual.*

Table 16-1. TACL System Security Features (page 2 of 2)

Command or Program	Function
REMOTEPASSWORD command	Runs the RPASSWRD program
RPASSWRD program	Establishes or deletes a password for a remote system
USERS program	Lists attributes for one or more users on the system
VOLUME command	Temporarily changes the default volume, subvolume, and file security, or resets these settings to their original default values
WHO command	Displays default information, including the file security attributes, for the current TACL process

For a detailed description of these commands and programs, see the *TACL Reference Manual.*

Table 16-2. FUP Disk-File Security Features

Command	Function
FUP GIVE	Changes the owner of a file
FUP INFO	Displays the characteristics of a file
FUP LICENSE	Lets nonprivileged users execute a privileged program (user ID must be 255,255)
FUP REVOKE	Revokes the license for a privileged program, or resets the security attributes of a file (user ID must be 255,255)
FUP SECURE	Changes the Guardian security attributes for a file

For a detailed description of these commands, see the *File Utility Program (FUP) Reference Manual.*

System Users

The system prevents access from unauthorized users. A group manager or super ID user assigns each user a unique user name and user ID to each user. To log on to systems that require passwords, a user must enter the user name or ID and the password.

NonStop™ Kernel system users fall into one of these classes, indicated by the user ID:

General users	Log on to a system to run one or more specific applications such as a text editor or manufacturing application. (General users are sometimes called application users.)
Group managers	Are responsible for members of a specific group on the system. The user ID is *n*,255, where *n* is the number of the group.
System operators	Perform various system functions such as managing system files, disks, and other devices. The user is ID 255,*n*, where *n* is an integer from 1 to 254.
Super ID users	Can access files, processes, and devices for the entire system with no restrictions. The user ID is 255,255.

Identifying System Users

Each system user has a unique user name and user ID in the form:

```
group-name.user-name
```

where *group-name* is the group to which the user belongs, and *user-name* identifies the individual user within the group.

A user ID is a pair of integers in the form:

```
group-id,user-id
```

where *group-id* identifies the user's group, and *user-id* identifies the user within the group. Each integer is in the range 0 through 255.

All user names and user IDs are kept in a system file. During a logon procedure, the system checks this file to ensure that the user name in the LOGON command is valid and that the correct password, if required, is supplied.

Capabilities of System Users

Each class of system users has different capabilities:

System User Class	Capabilities
General (or application) User	• Log on to the system to start a TACL process • Display system and user status • Assign and change a logon password • Set the default volume and subvolume names • Create, rename, and purge disk files • Set the security for disk files • List the names of all disk files • Run, debug, and stop processes • Start a backup TACL process and switch primary control to this backup process • List all groups and their users • Set remote passwords • Log off the system
Group Manager	• Perform all functions of a General User • Add new users to the group • Delete users from the group • Log on as any user in the group without knowing that user's password (the manager also has access to the user's files)
System Operator	• Perform all functions of a General User • Monitor processor use • Reload processor modules • Set the current date and time for the system • Alter bus availability states (hardware paths)
Super ID User	• Perform all functions of a General User, Group Manager, and System Operator • Add new groups to the system • Delete groups from the system • Add new users to the system in any group • Delete users from the system • Debug and stop any user's processes • Debug privileged programs • Log on as any user in any group without knowing that user's password

Adding New Users

When a new system is initialized, only these two users exist:

- A super ID user with the user name SUPER.SUPER and the user ID 255,255
- A null user with the user name NULL.NULL and the user ID 0,0

With the ADDUSER program, the super ID user creates new groups and adds new users to these groups. After being added by the super ID user, a group manager can also use the ADDUSER program to add new users to the group. For each new user, a user name and corresponding user ID must be specified.

For example, on a new system that was just initialized, the system manager (super ID user) can create a user group named ADMIN with group ID 6:

1. With the ADDUSER program, the system manager creates a group manager:

```
10> ADDUSER ADMIN.MANAGER, 6,255
```

2. The group manager can now add new users to the group. For example, the ADMIN.MANAGER user can now use the ADDUSER program to add these users to the group:

```
11> ADDUSER ADMIN.JOAN, 6,11
12> ADDUSER ADMIN.CHRIS, 6,12
13> ADDUSER ADMIN.JOHN, 6,13
14> ADDUSER ADMIN.NADINE, 6,14
15> ADDUSER ADMIN.MIKE, 6,1
```

   As many as 256 groups with a maximum of 256 users in each group can be created for each system.

3. The new user can log on with the TACL LOGON command to access the system:

```
1> LOGON ADMIN.JOAN
Password:
```

If ADMIN.JOAN has a password (which was supplied by the super ID user or group manager), the user enters this password at the “Password:” prompt. Otherwise, ADMIN.JOAN should press Return at this prompt. In either case, the user should use the PASSWORD program to change or select a password (provided the system requires a password).

The TACL program displays its prompt, which is a number and a greater-than sign (>). ADMIN.JANE can now enter TACL commands or run an application program.

For more information about logging on to the system and passwords, see Section 2, Getting Started With TACL.

Disk-File Security

Each disk file has an owner and a file security. When you create a file, you are its owner, and the file ownership is identified as your user ID. You remain the owner of the file unless you, your group manager, or a super ID user (255, 255) delete it or transfer its ownership to another user. As the file owner, you can secure the file with the File Utility Program (FUP) to prevent unauthorized users from accessing it.

When you create a file, your default logon security is automatically assigned to any file you create during a TACL session. To determine your default security, use the TACL WHO command. Unless you specify a different security for a file, all files that you create will have this default security.

Setting File Security

The four types of access for a disk file are read, write, execute, and purge (RWEP):

Table 16-3. Types of File Access

Access	Definition
Read	Lets a file be read or copied, and lets a command file be executed using the TACL OBEY command.
Write	Lets a file be modified.
Execute	Lets a file be executed as a process using the TACL RUN command (applies to program files with file code 100).
Purge	Lets a file be deleted or renamed, or to have its definition altered.

You set file security with the FUP SECURE command. You set your default security attributes with the TACL DEFAULT command or TACL VOLUME command. DEFAULT sets the logon (or saved) attributes, while VOLUME temporarily sets the attributes.

Table 16-4. Levels of Disk-File Security

FUP Code	Program Value	Access
–	7	Local super ID only
U	6	Local or remote owner (any user with the owner's user ID)
C	5	Local or remote member of the owner's group (any member of the owner's community)
N	4	Any local or remote user
O	2	Local owner only
G	1	Local member of the owner's group
A	0	Any local user

Local refers to access within a single system; remote refers to access between systems (or nodes) in a network.

For example, if you want to secure the file MYFILE as:

Read access	any local or remote user (N)
Write access	local owner only (O)
Execute access	any local or remote user (N)
Purge access	local owner only (O)

Enter this FUP command at the TACL prompt:

```
> FUP SECURE MYFILE, "NONO"
```

Each letter in NONO sets the respective RWEP attribute for MYFILE.

Accessing Disk Files

A user who accesses a disk file is classified as either a local or remote user. A local user is logged on to the system where the file resides; a remote user is logged on to a different system in the network. The security level of the user of a file is determined by:

User ID	Whether the opener is the owner of the file, a member of the owner's group, or a member of another group
Location	Whether the opener is local or remote with respect to the file

When you attempt to access a file, your security level is checked against the file's security level for the requested access mode (RWEP), as defined in Table 16-4.

Table 16-5 shows the permissions required for users to access files on local or remote nodes.

Table 16-5. Allowed Disk-File Access

If the user ID is	The user can access the file on the same node if the file has any of these permissions	The user can access the file on any node if the file has any of these permissions
Super ID	-, O, G, A	U, C, N
Owner or owner's group manager	O, G, A	U, C, N
Member of owner's group	G, A	C, N
Any user	A	N

For example, if a file owned by ADMIN.BILL is secured by the FUP SECURE command as follows:

```
-SECURE BILLFILE, "AGNU"
```

This security string means that any local user can read the file, only local members of the ADMIN group can write to the file, any network user can execute the file, and only the owner (whether logged on locally or remotely) can purge it.

If a second user ADMIN.ANN is on a remote system in the network, she can only execute the file. However, if she logs on locally, she also has read and write access for BILLFILE.

Default Security of Remote Files

The default security of a file that you create on a remote system might not be the same as your default security on your home system. For example, if your default security is GGGU, and you create a file on a remote system, its default file security is CCCU.

To give you, the owner of the file, the ability to access the file on the remote system, your local file access is converted to local and remote access for that remote file. This means that the local files access codes A, G, and O are converted to N, C, and U, respectively, for files you create remotely.

Process Security

The system can prevent one process from interfering with another process. Process security features, however, do not interfere with applications running on systems where security is not required.

This subsection describes the security features that protect and restrict access to running processes, such as process and creator access IDs, and describes their use in the security system. Also described are procedures for licensing programs and for adopting the user ID of a program file owner as that program's process access ID.

Process and Creator Access IDs

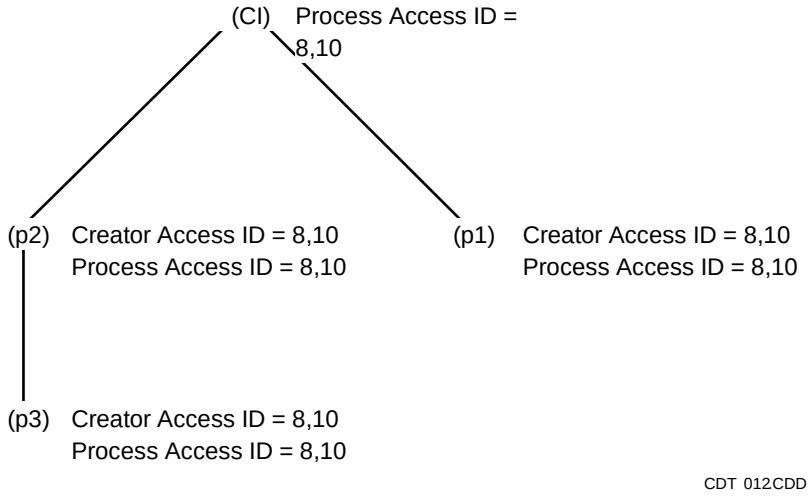
Two identifications are associated with each process: the creator access ID and the process access ID. The creator access ID identifies the user who initiated the creation of the process. The process access ID, which is often the same as the creator access ID, identifies the process and is used to determine if the process has the authority to make requests to the system (to open a file, stop another process, and so on).

Figure 16-1 shows a chain of process creations in which the process access ID of the original process is passed to other generations of processes. In this figure, CI is a command interpreter process, p1 and p2 are processes created by CI, and p3 is a process created by p2.

A process can determine its creator access ID and process access ID using the CREATORACCESSID and PROCESSACCESSID procedures, respectively (see the *Guardian Programmer's Guide*).

The process access ID is used to determine if file access is allowed. The process access ID is also used to determine whether certain security-restricted operations can be performed if the requester is neither the creator of the process nor the super ID.

Figure 16-1. Passing of Access IDs



ALTERPRIORITY, SUSPEND, STOP, and DEBUG are examples of security-restricted operations. If you need to perform security-restricted operations and do not have the appropriate process-access permissions, see your system administrator, security administrator, or group manager for assistance. For more information about security-restricted operations, see the *Guardian Procedure Calls Reference Manual*.

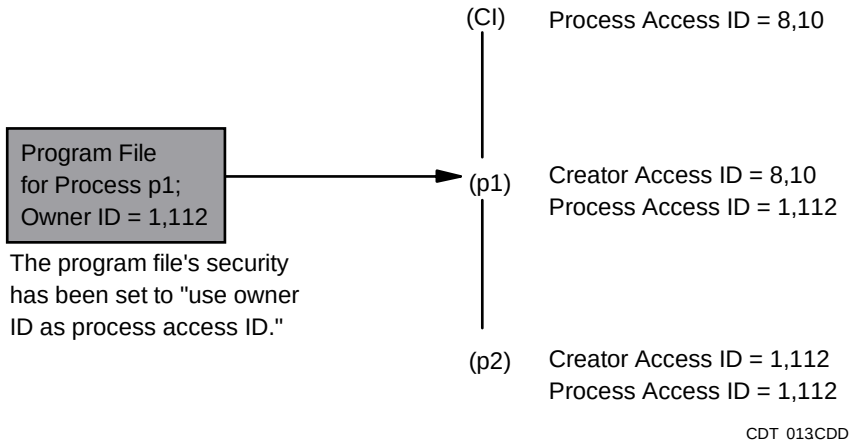
When a process is created, the operating system passes the appropriate process access ID to the descendant process. This ID becomes the creator access ID of the new process. The process access ID of the new process can come from either of two sources: the process access ID of its creator (this is the usual case), or the owner ID of the program file (if file adoption was specified with the FUP SECURE PROGID attribute).

Adopting the Owner ID of a Program File

Program file owner ID adoption allows the owner of a program file (or the super ID) to specify that the process access ID of any process created by running that program file is to be the same as the owner ID of the program file rather than the process access ID of the creating process. This option allows the owner of the program file to control the files that the new process can access and to control the operations that can be performed on or by the process. Program file ID adoption is specified with the FUP SECURE command (PROGID option) or the SETMODE or SETMODENOWAIT procedure.

Figure 16-2 shows several generations of processes and demonstrates how creator access and process access IDs can change when the PROGID attribute is set on. CI is a command interpreter process with process access ID 8,10. P1 is a process created by CI, and p2 is a process created by p1.

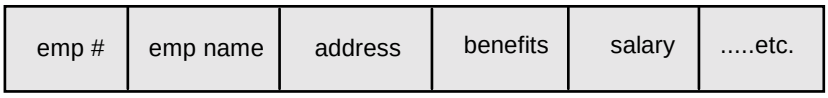
Figure 16-2. Effect of Adopting the Owner ID of a Program File



Controlled Access With Program File ID Adoption

In any application, some data files might require a controlled type of access—such as letting many users access certain records, while denying access to other records that are considered sensitive. For example, an employee file might contain such data as employees' identification numbers, names and addresses, and sensitive information such as salaries. This data might be in a record format as shown in Figure 16-3.

Figure 16-3. Employee Record Format

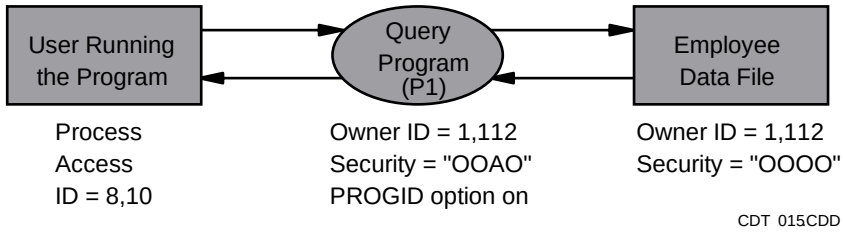


This example shows how a user can control the access to such a data file and also control any future file accesses or program functions.

An employee data file is owned by user 1,112 and is secured for local owner access only (OOOO). This means that only the file owner (or the local super ID) has direct access to the file. However, a controlled form of file access is allowed using a query program that has been written to return only nonsensitive information. The program file is owned by user 1,112 and is secured so that any local user can execute the process (OOAO). Additionally, program file ID adoption has been specified (use owner ID as process access ID).

As shown in Figure 16-4, user 8,10 (process access ID of 8,10) executes the query program, which returns "limited data views" only. The query process adopts the owner ID of the program file (1,112), which becomes its process access ID. (If the query program were to create another process, that process would inherit 1,112 as both its creator access ID and its process access ID.)

Figure 16-4. Controlled Access to a Data File



Licensing Programs

If a program contains privileged procedures (procedures having the CALLABLE or PRIV attribute), it must be licensed before it can be run by any user other than the super ID. Only a super ID user can license a file; licensing is performed with the FUP LICENSE command.

Programs running in the privileged mode have total freedom to access operating system tables and to execute privileged instructions and procedures, so it is possible for such programs to circumvent the file security checks and thereby gain access to any file. However, some privileged programs are needed in the system. Through licensing, the installation can run privileged programs that it has authorized, but users may not run unauthorized privileged programs. If a licensed file is opened with write access or read-write access, the file becomes unlicensed.

For example, a privileged program called PRIVPROG exists in a software development group. PRIVPROG is owned and licensed by the super ID so that all members of the group can execute it. A programmer in the group has developed a revision to the PRIVPROG program and wants to replace the object program with the revision.

Provided that the super ID user also gives the programmer write access to the program file, the following TAL compilation replaces the program with the revision and causes the program to become unlicensed:

```
10> RUN TAL / IN SOURCE / PRIVPROG
```

This means that no users except super ID users (not even the programmer who replaced the program) are allowed to execute the program. When PRIVPROG is debugged and ready for use, the super ID can license it so that others in the group can run it.

Network Security

This subsection describes several basic security tasks for systems connected on a network dealing with user access to files and processes.

Accessing a File on a Remote System

A user at system \WEST who wants to access a file (including a disk file, device, or process) on system \EAST must satisfy the following requirements:

- The user on system \WEST must also be established as a user on system \EAST.
- The user must have matching remote passwords established at both system \WEST and system \EAST.
- If the file is a disk file, the user on system \WEST must have the authority to access the file on system \EAST as a remote accessor.

Establishing Global User IDs

Each user is known to the local system by a user name and a user ID (for example, ADMIN.BILL and 6,14). A user can access files on a remote system only if the user's user name and user ID are also known to remote system.

For example, if ADMIN.BILL, who is on system \WEST, wants to access a file on remote system \EAST, the remote system must also have a user identified as ADMIN.BILL with a user ID of 6,14. A super group user (user ID 255,255) or a group manager at system \EAST must add ADMIN.BILL with the TACL ADDUSER program.

Establishing Remote Passwords

After user IDs for network users are added to relevant systems on the network, remote passwords must be established for each remote system. Remote passwords are specified with the TACL REMOTEPASSWORD command or the RPASSWRD program.

For example, ADMIN.BILL (user ID 6,14) was added at systems \WEST and \EAST. At system \WEST, these commands are entered to establish an allow-access remote password to system \WEST:

```
10> LOGON ADMIN.BILL



15> REMOTEPASSWORD \WEST, SHAZAM
```

The allow-access password for ADMIN.BILL for \WEST from all other systems is SHAZAM.

At system \EAST, these commands are entered:

```
10> LOGON ADMIN.BILL
11> REMOTEPASSWORD \WEST, SHAZAM
```

The user at system \EAST entered the matching password and now has remote access to system \WEST as ADMIN.BILL.

ADMIN.BILL, logged on at system \EAST, does not have the same status at \WEST as does the ADMIN.BILL at \WEST. Because ADMIN.BILL at \EAST is a remote accessor of \WEST, this user cannot access disk files on \WEST that are secured for local access only.

Also, if ADMIN.BILL on \EAST creates a process on \WEST that attempts to access the home terminal on \EAST, the attempt will fail because remote passwords to allow access from \WEST to \EAST were not established.

For ADMIN.BILL to gain access to \EAST from \WEST, an allow-access password must be defined for ADMIN.BILL at \EAST, matched by a request-access password at \WEST. For example, this is entered first at \EAST and then at \WEST:

```
10> LOGON ADMIN.BILL
11> REMOTEPASSWORD \EAST, AARDVARK
```

Now users logged on as ADMIN.BILL at either system \WEST or system \EAST have access to both systems.

Considerations for Remote Passwords

- When matching remote passwords are established on both systems, a user does not need to specify the remote password to gain access to the remote system. Furthermore, the super IDs at the various nodes in a network can set up the appropriate allow-access and request-access passwords for all users so that the users themselves need not be concerned with REMOTEPASSWORD commands. When the appropriate passwords are established for a user, the user can access files remotely without being aware of the network passwords.
- The absence of an allow-access password prevents remote access by anyone acting as that user. Thus, if MARKETNG.SUE does not supply an allow-access password, no remote user with the same user ID can access MARKETNG.SUE's home system as MARKETNG.SUE.
- A remote password, once defined, remains in effect until modified by a subsequent REMOTEPASSWORD command. To remove the remote password for the system \EAST:

```
10> REMOTEPASSWORD \EAST
```

  To remove all of the user's remote passwords:

```
10> REMOTEPASSWORD
```

- Request-access passwords and allow-access passwords can be specified at any time. Remote access is permitted as soon as both remote passwords are defined (provided they match).

- Remote passwords are independent of local passwords. In the preceding example, ADMIN.BILL could prevent unauthorized persons from logging on as ADMIN.BILL by entering this command with password LOCBILL at either system:

```
10> PASSWORD LOCBILL
```

Accessing Processes on a Remote System

Security Considerations for Remote Processes

- With respect to a specific system, each process in the network is either local or remote. A process is remote to a system if:
  - It is running on a remote system.
  - Its creator is on a remote system.
  - Its creator is remote.

  Therefore, a process that is running on a system can be remote with respect to that system. These restrictions prevent a remote process from creating another process to access a file whose security specifies local access only.
- A remote process cannot suspend or activate a local process. A remote process cannot stop a local process, unless the stop mode of the local process is 0 (which allows anyone to stop it).
- A remote process cannot put a local process into a debug state.

Using a Remote TACL Process to Gain Local Access

Openers of a file are either local or remote with respect to the file. A local user is logged on to the system on which the file resides. A remote user is logged on to a different system in the same network.

A remote accessor of a system can become a local accessor by running a TACL process in the remote system and logging on. For example, if remote passwords are established so that user ADMIN.BILL at \WEST can access system \EAST, ADMIN.BILL can issue these commands:

```
10> \EAST.TACL
TACL 1> LOGON ADMIN.BILL
Password:
```

ADMIN.BILL is now logged on as the local ADMIN.BILL on system \EAST. Therefore, ADMIN.BILL can access disk files on \EAST owned by ADMIN.BILL even if they are secured OOOO (local owner only) as well as other files that are only accessible locally.

This remote session can be terminated with the TACL EXIT or LOGOFF command, or with CTRL-Y. If ADMIN.BILL terminates the TACL process (identified by process ID

2,10) on system \EAST with an EXIT command or with CTRL-Y, the TACL program displays the message:

```
Are you sure you want to stop this TACL   (\EAST.2,10)?
```

Reply YES (or Y) to stop the remote TACL process.

If ADMIN.BILL enters a LOGOFF command, the TACL program terminates and displays this message:

```
Exiting from TACL on \EAST
```

A remote user can be prevented from becoming a local user if the local super ID specifies A (any local user) as the execute security for the TACL program file. This prevents anyone on a remote system from starting a TACL process on the local system.

Also, a user who has the same user name as a user in another system cannot log on to that system without knowing the local password for that user name. For example, ADMIN.BILL on system \WEST cannot log onto system \EAST if ADMIN.BILL at \EAST has a local password that is unknown to ADMIN.BILL at \WEST.

Establishing a Global Remote Password

In some networks, it is not preferable for all users to have access to all systems. However, it is convenient to allow network access for certain users without forcing them to type or even know all of the required REMOTEPASSWORD commands. In this case, a global remote password can be established for these users.

At each system, a user named NET.ACCESS is established and these commands are issued:

```
10> LOGON NET.ACCESS
12> PASSWORD local-password
13> REMOTEPASSWORD \WEST, global-password
14> REMOTEPASSWORD \EAST, global-password
15> REMOTEPASSWORD \NYNY, global-password



18> REMOTEPASSWORD \system-n, global-password
```

The REMOTEPASSWORD command is issued for each system on the network. The global remote password is the same for all systems and is known only to the system managers. The local password is different for each system and is given only to users who are allowed to access all systems on the network.

Only the users who know the local password can log on as NET.ACCESS. While logged on as NET.ACCESS, these users can access remote files. For example, this command lets users access remote files secured for access by NET.ACCESS.

```
1> LOGON NET.ACCESS, local-password
```

Establishing Subnetworks

In a large network, it is sometimes preferable to allow users to access some nodes but not others. For example, users on system \SANFRAN are allowed to access systems \LA, \SEATTLE, and \CUPRTNO but not the \NEWYORK and \CHICAGO systems.

In this case, the preceding examples can be extended to allow access to any number of subnetworks (that is, any collection of individual nodes). A user such as NET.WEST is established at each node of the subnetwork, and a password scheme like the one used in the previous example allows certain users to log on as NET.WEST.

Subnetworks implemented in this manner can overlap or include one another. \CHICAGO might be accessible from \NEWYORK by logging on as NET.EAST, and from \PHOENIX by logging on as NET.MIDWEST. Similarly, each system in the network might have a user called NET.GLOBAL, who is allowed to access every other node.

Capabilities of a Remote Super ID User

On a single system, a super ID user can access any file. On a network, the capabilities of the super ID can be local, global, or somewhere in between local and global.

Making the Super ID a Local Super ID Only

To make the super ID exclusively a local super ID user, do not issue REMOTEPASSWORD commands for the super ID at any node.

Making the Super ID a Global Super ID

To make the super ID a global super ID, issue REMOTEPASSWORD commands (as defined in Establishing a Global Remote Password on page 16-22) at every node, and give every super ID the same password.

In this case, if a disk file is secured A, G, O, or –, a remote super ID user can still gain access to the file by running the TACL program on that system and logging on as the local super ID.

Making the Super ID Between Local and Global

To make the super ID somewhere between a local and global super ID user, issue REMOTEPASSWORD commands (as defined in Establishing a Global Remote Password on page 16-22) at every node, but give each super ID a distinct password.

Thus, any disk file can be protected from remote access by giving it A, G, O, or – security. (The remote super ID can then access files secured N, C, or U.) A remote super ID cannot log on as a local super ID user because the password for the local super ID is unknown.

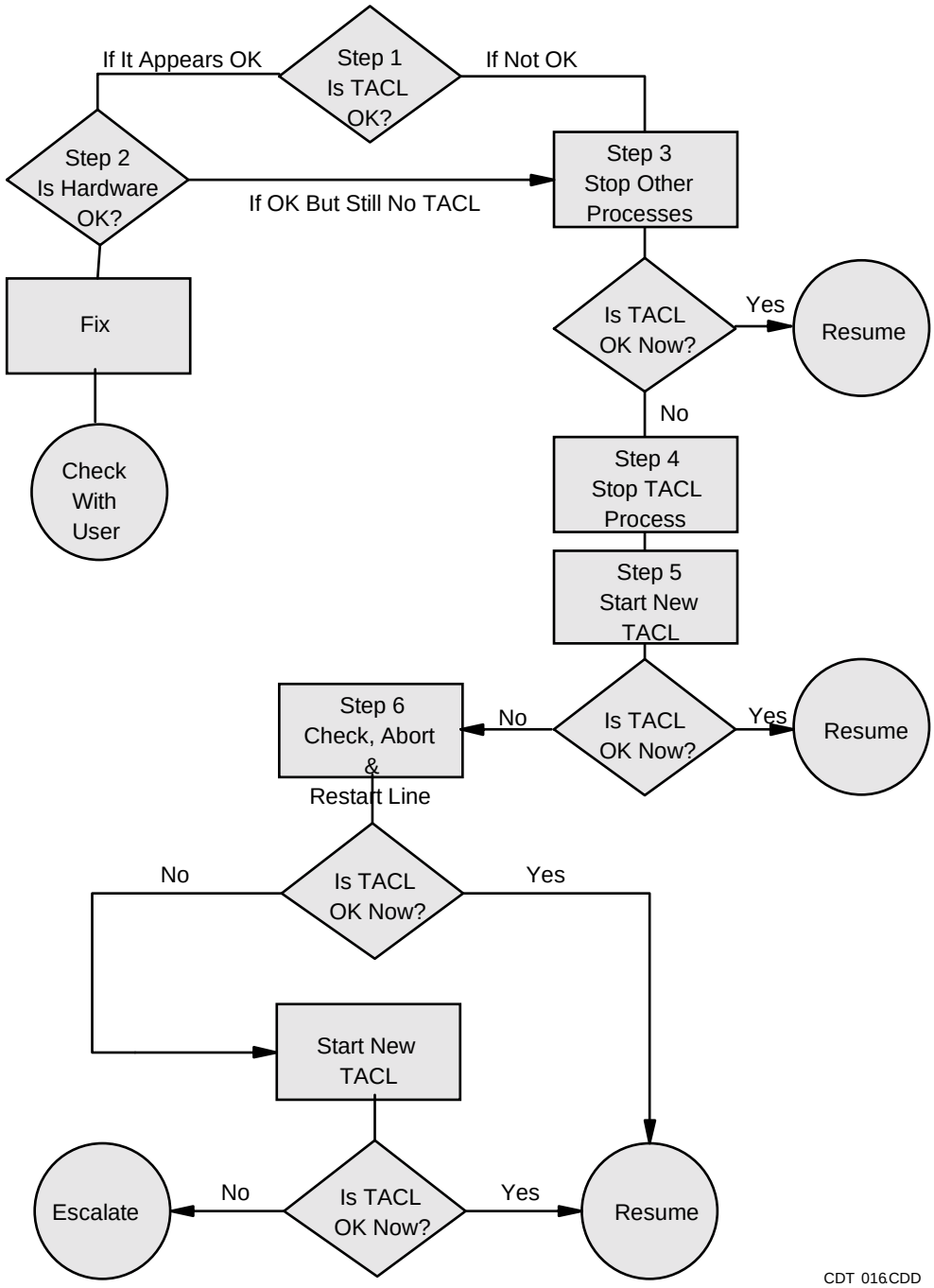
Solving System Access Problems

Sometimes users lose access to the TACL command interpreter. When they do, there are six simple tasks to perform that can help you easily restore their access:

The final topic listed above provides a table of common system access problems and their solutions.

This subsection applies only to users who have TACL access. Depending on your system configuration, some users might use other means for system access. Check with your system manager if you have questions about user access.

Figure 16-5. Flow Chart: Access Problem Troubleshooting Procedure



Task 1: Check the Status of the User's TACL Process

Check the status of a user's TACL process to see what processes are running on the user's terminal:

1. Determine the user's terminal name (for example $JT1.#J01) by asking the user or by checking your files or records, and make a note of it.
2. Check the status of all processes (including the TACL process):

   ```
   > STATUS *, TERM $terminal-name
   ```

If no TACL process is running, the user might have exited from the TACL process. If this is the case, go to Task 5: Start a New TACL Process on page 16-29. Otherwise, continue with Task 2: Check for Hardware Problems on page 16-27.

For more information about the STATUS command, see the *TACL Reference Manual*.

Example

To check the status of the TACL process running on the terminal $JT1.#J01, enter:

```
> STATUS *, TERM $JT1.#J01
```

Information such as this is displayed:

```
Process          Pri PFR %WT Userid  Program file               Hometerm
$TH02    4,100   150     005   8,001 $SYSTEM.SYS00.TACL         $JT1.#J01
$TH02 B  5,100   150     001   8,001 $SYSTEM.SYS00.TACL         $JT1.#J01
```

This status report example shows a normally functioning TACL process and includes:

- The process name, if it is named (Process)
- Whether the process is a backup or primary process (B if a backup process)
- The processor and process number (for example, 4,100; 5,100)
- The execution priority of the process (Pri)
- PFR code (PFR)
  - P indicates that the process contains privileged code.
  - F indicates that the process is waiting on a page fault.
  - R indicates that the process is on the ready list.
- The wait state (%WT)
- The group and user ID numbers of the person using the process (Userid)
- The name of the program file (Program file)
- The home terminal of the process (Hometerm)

Task 2: Check for Hardware Problems

For information on running hardware diagnostics, see the *TSM Configuration Guide* and the *TSM Online User's Guide.*

If system access problems continue, proceed to Task 3: Stop the User's TACL Process on page 16-27.

Task 3: Stop the User's TACL Process

1. Make sure you are logged on as the Super ID (255, 255) or as the user's group manager (*n*, 255).
2. Determine the name of the user's TACL process:

   ```
   > STATUS *, TERM $terminal-name
   ```

3. Note the TACL process name and the numbers of the primary and backup CPUs the TACL process is using. You will need these numbers later when you start a new TACL process.
4. Stop the TACL process by using the TACL STOP command:

   ```
   > STOP $TACL-process-name
   ```

See the *TACL Reference Manual* for complete syntax, considerations, and examples of the STOP command.

Example

To stop the TACL process that the user SALES.BONNIE is running on her terminal $JT1.#J01:

```
> STATUS *, TERM $JT1.#J01
```

```
Process          Pri PFR %WT Userid  Program file                Hometerm
$TH02    4,100   150     005   8,001 $SYSTEM.SYS00.TACL          $JT1.#J01
$TH02 B  5,100   150     001   8,001 $SYSTEM.SYS00.TACL          $JT1.#J01
         9,168   150     005   8,001 $SYSTEM.SYSTEM.TEDIT        $JT1.#J01
```

```
> STOP $TH02
```

Task 4: Stop Other User Processes

Any user processes you stop in this step can be restarted once system access is restored. If there are no other user processes to stop, proceed to Task 5: Start a New TACL Process on page 16-29.

1. Make sure you are logged on as the Super ID (255, 255) or as the user's group manager (*n*, 255).
2. Determine the names of the user's running processes by entering:

   ```
   > STATUS *, TERM $terminal-name
   ```

3. Note the names of any running processes. For unnamed processes, note the CPU and PIN numbers of those processes. You will need them to stop the unnamed processes.
4. Stop user processes either individually or all at once. To stop user processes one at a time, go to Step 4a. To step all running processes at once, go to Step 4b.
   a. You might want to stop any running processes individually to ensure there are no problems, or if there is a certain order in which they must be stopped.
      1. Stop a named process by using the TACL STOP command:

         ```
         > STOP $process-name
         ```

         Repeat this command for each named process you need to stop.
      2. Stop an unnamed process by using the TACL STOP command:

         ```
         > STOP cpu,pin
         ```

         Repeat this command for each unnamed process you want to stop.

   b. Stop all running processes except the user's TACL process in one command by entering the following command from the user's terminal (if entered from another terminal, this command stops everything on that user's terminal):

      ```
      > STATUS *, TERM $terminal-name, STOP
      ```

See the *TACL Reference Manual* for complete syntax, considerations, and examples of the STOP command.

Example

To stop the unnamed TEDIT process that the user SALES.BONNIE is running on her terminal $JT1.#J01, enter:

```
> STATUS *, TERM $JT1.#J01
```

```
Process          Pri PFR %WT Userid  Program file                Hometerm
$TH02    4,100   150     005   8,001 $SYSTEM.SYS00.TACL          $JT1.#J01
$TH02 B  5,100   150     001   8,001 $SYSTEM.SYS00.TACL          $JT1.#J01
         9,168   150     005   8,001 $SYSTEM.SYSTEM.TEDIT        $JT1.#J01
```

```
> STOP 9,168
```

The specified process is stopped.

Task 5: Start a New TACL Process

If there are no obvious problems indicated in Tasks 1 through 4, starting a new TACL process is the next task to try for restoring system access.

1. Using the TACL process name and the CPU numbers you noted in Task 3, start a new TACL process:

   ```
   > TACL /IN $terminal-name, OUT $terminal-name, NAME
     $process-name, NOWAIT, PRI nn, CPU primary/backup
   ```

   If you enter NAME followed by a space and a comma, the operating system assigns a unique name to the new TACL process. However, it is best to use the name of the old TACL process, because that name is usually included in a system configuration file that assigns names to TACL processes when the system is loaded the next and subsequent times.

2. Check the status of the new TACL process:

   ```
   > STATUS *, TERM $terminal-name
   ```

   The system displays whether the new TACL process has started on the user's terminal.

3. Check with the user to make certain their access to the TACL command interpreter has resumed.

   - If access is restored, instruct the user to restart any processes that were stopped in Task 4.
   - If access is still not restored, proceed to Task 6: Check, Stop, and Restart Terminal Communication Lines on page 16-30.

If you are restarting a TACL process, see Restarting a TACL Process on page 4-9 for more information.

Example

To start a new TACL process for the user SALES.BONNIE on the terminal $JT1.#J01, enter:

```
> TACL /IN $JT1.#J01, OUT $JT1.#J01, NAME $TH02, NOWAIT, PRI
150,&
  CPU 4/5

> STATUS *, TERM $JT1.#J01
```

Task 6: Check, Stop, and Restart Terminal Communication Lines

If a user still has no TACL access after you have performed Tasks 1 through 5 of this subsection, you should check and possibly stop and restart the user's terminal communication line.

1. Check the user's terminal communication line.
   a. Make sure you are logged on as a super-group user (255,*n*).
   b. Run the SCF utility.

      ```
      4> SCF
      ```

   c. At the SCF prompt, ssume the user's terminal line name:

      ```
      -> ASSUME LINE $line
      ```

      line is the terminal line name, and is the same as the first component of the user's terminal name; for example, $JT1.

   d. Assume the subdevice name of the user's terminal:

      ```
      -> ASSUME SU #subdevice
      ```

      subdevice is the subdevice name, and is the same as the second component of the user's terminal name; for example, #J01.

   e. Check the status of the line and subdevice:

      ```
      -> STATUS
      ```

      The STATE display appears indicating whether the terminal communication line is started or stopped.

      - If the STATUS command reports STARTED, the terminal communication line is working. You should follow your local procedure for problem reporting and escalation.
      - If the STATUS command reports STOPPED, the terminal communication line is not working. Proceed to Step 2.

2. Stop the terminal communication line and subdevice.

   ```
   -> ABORT
   ```

Caution. The ABORT command will stop the subunit #*subdevice* when a subunit is assumed with the ASSUME command. Aborting a line will usually affect other users connected to subunits associated with the line, so it is not usually done unless all subunits (subdevices) on a particular line are being affected by a problem.

---

3. Restart the line and subdevice.

   ```
   -> START
   ```

4. Check the status of the line and subdevice.

   ```
   -> STATUS
   ```

   The STATE display appears indicating whether the terminal communication line is started or stopped.

   - If the STATUS command reports STARTED, the terminal communication line is working.

     1. Exit SCF.

        ```
        -> EXIT
        ```

     2. Restart a TACL process for this line by following Task 5: Start a New TACL Process on page 16-29. If you still cannot start a TACL process, follow your local procedure for problem reporting and escalation.

   - If the STATUS command reports STOPPED, the terminal communication line is not working; follow your local procedure for problem reporting and escalation. See the *SCF Reference Manual for G-Series Releases* for complete syntax, considerations, and examples of all SCF commands.

See Section 19, Monitoring Hardware Components, for information on checking for LAN problems.

Example

1. Check the status of the terminal line and subdevice $JT1.#J01.

   ```
   1> SCF
   -> ASSUME LINE $JT1
   -> ASSUME SU #J01
   -> STATUS
   ```

   When there is a problem with the line, the status shows that the state of the line is STOPPED.

2. Stop the subunit.

   ```
   -> ABORT
   ```

3. Restart the subunit (subdevice).

   ```
   -> START
   ```

4. Check the status of the subunit (subdevice).

   ```
   -> STATUS
   ```

   When the line is functional, the status shows that the state of the line is STARTED.

5. Exit SCF.

   ```
   -> EXIT
   ```

Common Terminal and Workstation Problems

Table 16-6 lists possible problems, causes, and solutions for common problems that can occur with terminals and workstations.

Table 16-6. Common Terminal and Workstation Problems

Problem	Possible Causes	Solution
A terminal or workstation is unresponsive.	A terminal hardware problem might exist.	Turn the power off and on again. If a self-test fails, repair or replace device.
	A terminal configuration problem might exist.	Reset configuration switches according to manufacturer's instructions.
	Terminal cabling might be faulty.	Tighten or replace the cable.
	A modem might be faulty.	Repair or replace the modem in question.
	Communication lines might be faulty.	Attempt to restart line. Escalate as needed.
	A controller problem might exist.	Check for error messages.
	A TACL problem might exist.	Restart the TACL process.
	A user process has failed.	Stop and restart the process.
	A terminal emulator software configuration problem might exist.	Check manufacturer's documentation or escalate the problem according to local procedure.
	A local area network hardware problem might exist.	Check manufacturer's documentation or escalate the problem according to local procedure.
	A local area network software problem might exist.	Check manufacturer's documentation or escalate the problem according to local procedure.
Performance is poor	The communication line might be overloaded or the line error rate is high.	Check line status with SCF, stop the line, and restart it if indicated. If problems persist, escalate the problem according to local procedure.

17 Monitoring Event Messages

Operator messages are often related to hardware malfunctions or to important changes in the state of a system component. They help you ensure that all of your system components are up and running.

Regularly monitoring operator messages is an important part of your job. Many companies recommend that operations staff monitor messages every fifteen to twenty minutes throughout the day. Frequent monitoring of messages can help you detect potential problems at an early point and lets you take preventive or corrective measures.

Understanding Operator Messages

When a subsystem or component of the Compaq *NonStop*™ Kernel operating system detects an event that might affect its operation, it generates a message that describes the event. When an event message is intended to be seen, and perhaps acted upon, it is converted into displayable text and called an operator message.

Operator messages include system-generated and user-generated messages. System-generated messages are produced by the NonStop™ Kernel operating system and by NonStop™ Kernel subsystems as they monitor and manage system resources. User-generated messages are produced by user-written applications and are not covered in this guide. You should be familiar with this operator message terminology:

Messages Term	Definition
Event message	A message sent by the operating system or a reporting application that describes an event that occurred in a system or network. They do not appear in a readable format; they are converted into operator messages.
Operator message	Event message that has been converted into readable text. Operator messages can be displayed in different formats, depending on the application that formats them.
Collector	Event Management Service (EMS) process that accepts event messages from subsystems and logs them in an event log.
Event log	File or set of files maintained by EMS to store event messages generated by various subsystems.
Distributor process	EMS process that distributes event messages from event logs to requesting management applications, another collector on this or another node, or printers, devices, or files. EMS provides several distributors: ● Consumer distributor ● Forwarding distributor ● Printing distributor ● Compatibility distributor ($Z0)
Consumer distributor	EMS distributor that returns selected event messages to management applications upon request. The distributor is used by NonStop™ Kernel applications (such as TSM EMS Event Viewer) and by users to read EMS or alternate log files and all or specific event messages, depending on filter specifications loaded to the distributor. The application can take appropriate action, if necessary, in response to important events.
Forwarding distributor	EMS distributor that sends selected event messages to an EMS collector on another network node or, if the node has multiple collectors, to an EMS collector on the same node.
Printing distributor	EMS distributor that sends operator messages to printers, devices, processes, or files.
Compatibility distributor ($Z0)	Allows downward compatibility with a previous operator process. The compatibility distributor sends operator messages to a console device during system load.

Operator Message Monitoring Tools

Several utilities on your system can help you monitor operator messages, including:

- **Event Management Service (EMS)** provides event-message collection, logging, and distribution facilities for the NonStop™ Kernel operating system. For more information, see the *EMS Manual*.
- **TSM EMS Event Viewer** helps you perform many of the tasks associated with viewing and monitoring EMS event logs ($0 and $ZLOG). It enables you to search for and view the log files in a variety of ways and retrieve events based on start and end time, subsystem, source, and multiple or specific events. For more information, see the *TSM Online User's Guide* and the TSM EMS Event Viewer Application online help.

Operator Message Types

Message Type	Definition
Common	Messages "common" to several subsystems; they have negative event numbers.
EMS	Messages sent under the Event Management Service (EMS) subsystem ID.
OSS	Messages sent under the Open System Services (OSS) subsystem ID.
Subsystem	DSM display format messages generated by a specific subsystem.

For more information, see the *Operator Messages Manual*.

Operator Messages Format

Operator messages are preceded by a header that contains:

- Subsystem ID
- Event number
- Time the message was generated
- Name of the system that sent the message
- Process ID or processor number of the process that issued the message

Only portions of the subsystem ID and the event number are shown as header information in this guide.

This example shows the format for Compaq Open Systems Interconnection/Message Handling System (OSI/MHS) operator messages as they are sent to disk files, terminals, or printers:

```
96-10-03 15:09:51 \NET.$FTI1     TANDEM.MHS.G01                 000023 $ZL1
                                   MR.\NET.MRGRP1  $RL11: OSI resource
                                   problem with device \NET.$LAPX.#Z0004BW
                                   on call APS_ASSOC_CONNECTREQ  returned
                                   with -1001, 140
```

How Operator Messages Are Created

The Event Management Service (EMS) is a set of processes that collects event messages from reporting processes and subsystems. The EMS primary collector process ($0) is the primary collection point for these event messages. $0 collects messages generated by reporting subsystems and writes these messages to an event log file.

The event log file is read by the EMS distributor processes configured onto or started on your system. The distributor processes then send messages to various destinations.

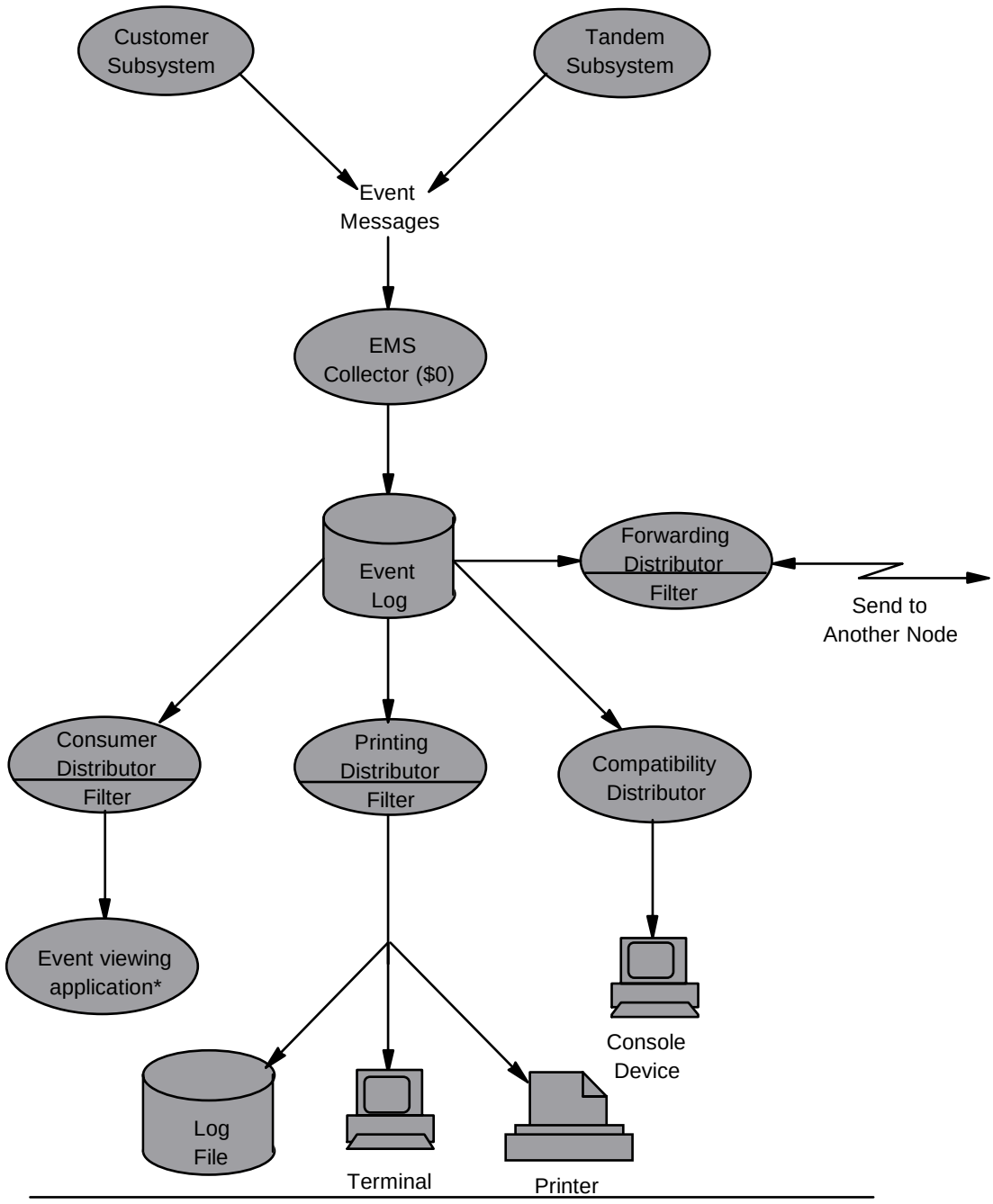
Table 17-1. Distributor Processes and Message Destinations

This Distributor Process:	Sends These Messages to These Destinations:
Consumer distributor	Sends event messages to management applications
Printing distributor	Sends operator messages to disk files, terminals, or printers
Forwarding distributor	Sends event messages to another node on your network
Compatibility distributor ($Z0)	Sends operator messages to a console device for EMS message monitoring during system loads

The $Z0 compatibility distributor can be configured only in EMS mode. It formats event messages from the event log file into operator messages and distributes them to the local operator console.

Figure 17-1 shows these EMS components and how they interact to route operator messages. The *EMS Manual* describes these processes in detail.

Figure 17-1. Operator Messages and the EMS Environment



Displaying Error Messages With Error

When you are using the TACL program or a Guardian utility, you can receive file-system error numbers at your terminal. You can use the Error program to display an explanation of the error number.

Running Error

You run Error from the TACL program; the syntax is:

```
ERROR [ / run-option / ] error-number
```

run-option

Any run option described under the RUN command in the *TACL Reference Manual*.

error-number

A file-system error number. If you enter an invalid error number, Error displays an error message. If you specify -1, Error displays all error-number descriptions. To stop displaying the list of all error-number descriptions, press the Break key and enter STOP.

Error Examples

This example shows a FUP DUPLICATE command that generates file-system error 11:

```
30> FUP DUP OLDFILE, NEWFILE
ERROR - $DISK2.COMPILES.OLDFILE: ERR 11
*ABEND*
ABENDED: 15,47
```

FUP displays file-system error 11 and stops (the FUP process ID is 15,47). For an explanation of this error, enter Error, a space, and the error number at your TACL prompt:

```
31> ERROR 11
0011 The file is not in the directory or the record is not in the file, or the
     specified tape file is not on a labeled tape.
```

Error displays the error description and returns control to the TACL program. From the description, you can determine that OLDFILE is not a valid file. You can then correct the file name and reexecute the FUP DUPLICATE command.

This example shows a file-system error generated by a TACL PURGE command:

```
40> PURGE FILE89
$ACCTS.CUSTOMER.FILE89 File error 48
```

Use Error to display the explanation of the error:

```
41> ERROR 48
0048 Security violation; illegal operation attempted.
```

A security violation occurred; you cannot purge FILE89 because you don't have purge access for the file.

Displaying Operator Messages With a Printing Distributor

You can display operator messages by setting up a printing distributor. This subsection describes how to start and stop a printing distributor, and how to configure it to begin and stop at times you specify.

Note. Normally a dedicated terminal is used to display operator messages. Using a dedicated terminal, which displays messages at all times, usually makes it unnecessary for you to start and stop a printing distributor.

Starting a Printing Distributor

To start a printing distributor and have it send operator messages to your terminal, disk file, or printer, enter this command at your TACL prompt:

```
> EMSDIST TYPE PRINTING, COLLECTOR $0, TEXTOUT $device
```

where $*device* is the name of the terminal, disk file, or printer to which you want to send operator messages.

The printing distributor then begins displaying or directing operator messages to the device you have specified, as in this example:

```
96-10-03 15:09:51 \NET.$FTI1     TANDEM.MHS.G01                 000023 $ZL1
                                   MR.\NET.MRGRP1  $RL11: OSI resource
                                   problem with device \NET.$LAPX.#Z0004BW
                                   on call APS_ASSOC_CONNECTREQ  returned
                                   with -1001, 140
```

Stopping a Printing Distributor

1. Press the Break key on your keyboard.
2. Enter the TACL STOP command with the name of your EMSDIST process. To learn the name of your EMSDIST process, enter:

   ```
   > STATUS *, TERM
   ```

3. Search for the EMSDIST process in the left-hand column of the STATUS display. Then enter the STOP command:

   ```
   > STOP $process-name
   ```

You can specify that a printing distributor begin or end or both on a specified date at a specified time by using the EMSDIST TIME and STOP options. See the following example and to the *EMS Manual* for instructions on using these options.

Example

This example uses the TIME and STOP options of the EMSDIST command to:

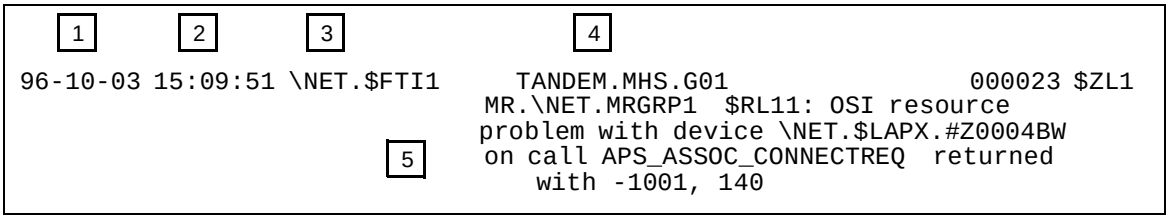
- Begin a printing distributor immediately
- Print operator messages starting at 1:00 a.m. on May 12, 1993
- Print the messages on the printer associated with spooler location #HANS1
- Stop the distributor and the printing of messages at 11:00 p.m. the same day

```
> EMSDIST TYPE PRINTING, COLLECTOR $0, TEXTOUT $S.#HANS1,&
> & TIME 1993-05-12 1:00:00, STOP 1993-05-12 23:00:00
```

Interpreting Operator Messages

Operator messages can be configured to appear in different formats, depending on the way your system has been configured. For example, you might use a printing distributor, console application, or other application to display operator messages.

When you use a printing distributor to display operator messages, they appear in a format similar to this example:



1 the timestamp

2 the names of the system and the process reporting this message

3 the owner name, subsystem name, and release number

4 the event number

5 the text of the message

To interpret the message shown above:

1. Determine the subsystem name and event number from the information in the operator message header. In this example, fields 3 and 4 contain the information you need.

   MHS is the subsystem name and 000023 is the event number.

2. Turn to the appropriate section in the *Operator Messages Manual* (in this example, it is the section describing MHS messages).

3. Search for the event number. The probable cause, effect, and suggested recovery steps are included for each event number.

The message's subsystem ID and event number variables include:

owner-name	TANDEM for all messages described in this guide.
subsystem-name	Subsystem's name. In rare cases, the subsystem number instead.
version	Release number (for example, G06).
event-number	Event number.

Examples

TSM EMS Event Viewer Message Format

For operator messages displayed using the TSM EMS Event Viewer application, the message's subsystem ID and event number are displayed on the TSM EMS Event Viewer main window. If you see events without a subsystem ID, subsystem name, and event number, see the *Operator Messages Manual* to find out what the message means. The main window of the TSM EMS Event Viewer displays messages in this format:

```
Date      Time      SSID              Subject  Event # Event Name   Event Text

10/03/96 08:10:23 Tandem.DSK.G01  $SYSTEM  500     Path-Switch  Operator's
                                                                 Console
                                                                 Message
```

The SSID column identifies the *owner-name*, *subsystem-name*, and *version*. The Event # column displays the *event-number*.

Printing Distributor Message Format

For event messages sent to printers, log files, and terminals used by the printing distributor, the message's subsystem ID and event number appear in this format:

```
owner-name.subsystem-name.version          event-number
```

For example:

```
TANDEM.TMF.G06        000041
```

Directing Messages to a Disk File

One way operations staff can monitor operator messages is to direct messages to a disk file. This file can then be printed when you want: see Printing Operator Messages on page 17-11 for instructions on how to print messages from a disk file. You can also view the file at your terminal if you want to.

To start a printing distributor and use it to direct operator messages to a disk file for future printing or perusal, enter:

```
> EMSDIST TYPE PRINTING, COLLECTOR $0, TEXTOUT &
> & $disk.subvol.file
```

Messages are then sent to the specified disk file until you stop the EMSDIST process with the TACL STOP command.

Example

This example shows how to direct operator messages to a specific file then display them on your terminal.

1. Direct operator messages to the disk $OPS, subvolume LOGS, and file MSG1:

   ```
   > EMSDIST TYPE PRINTING, COLLECTOR $0, TEXTOUT $OPS.LOGS.MSG1
   ```

   Disk file logging begins and continues until you stop the printing distributor.

2. When you want to return to a TACL prompt, press the Break key on your terminal.

3. Display the contents of the disk file MSG1 on your home terminal:

   ```
   > FUP COPY LOGS.MSG1,, SHARE
   ```

   The SHARE option of the FUP COPY command, which must include two commas as shown, is necessary because the log file is likely to be open at any time.

Note. You can specify that the logging of messages to your disk file begin, end, or both at a specified date and time by using the EMSDIST options TIME and STOP. For an example of using these options, see Printing Operator Messages on page 17-11. See the *EMS Manual* for complete information about EMS, its options, and for instructions on how to use them.

Printing Operator Messages

In addition to displaying operator messages using a print distributor, you can also print operator messages using a printing distributor and the FUP COPY command.

To use a printing distributor to print operator messages:

1. Supply the name of an existing print device in an EMSDIST command.

   ```
   > EMSDIST TYPE PRINTING, COLLECTOR $0, TEXTOUT $printer.#loc
   ```

   The printing distributor begins and continues to direct operator messages to the specified printer until you stop the EMSDIST process.

   (These instructions assume that you have designated a disk file for operator messages. If you need to do so, see Directing Messages to a Disk File on page 17-10.)

2. After a disk file exists, print the operator messages with the FUP COPY command:

   ```
   > FUP COPY [subvol.]log-file, $printer.#loc, SHARE
   ```

   You must use the SHARE option of the FUP COPY command to print a log file because this type of file is likely to be open at any time.

Note. See the *EMS Manual* for complete information about EMS, its options, and for instructions on how to use them.

Example

To print the contents of the disk log file NEWLOG to a printer named #HANS1, enter:

```
> FUP COPY NEWLOG, $S.#HANS1, SHARE
```

You can specify that logging begin or end or both on a specified date at a specified time by using the EMSDIST TIME and STOP options. This example begins the printing distributor immediately, starts printing at 1:00 a.m. on May 12, 1993, on the printer named #HANS1, and stops the distributor and the printing of messages at 11:00 p.m. the same day:

```
> EMSDIST TYPE PRINTING, COLLECTOR $0, TEXTOUT $S.#HANS1,&
> & TIME 1993-05-12 1:00:00, STOP 1993-05-12 23:00:00
```

Guidelines

- When you print a log file with the FUP COPY command, you must use the SHARE option if the file is currently open, which is often the case. For example:

  ```
  > FUP COPY NEWLOG, $S.#HANS1, SHARE
  ```

- The person or process that runs EMSDIST must have read access to the log files that EMSDIST accesses. Super-group privileges are required if the collector creates its own log files with the protection string COOO, which is the system default file security. See the *EMS Manual* for further information.

Monitoring Messages With the TSM EMS Event Viewer

The TSM EMS Event Viewer (Event Viewer) assists you in performing many of the tasks associated with viewing and monitoring various EMS event logs. The Event Viewer lets you set up criteria to view the log file in several ways, enabling a rapid assessment of service problems.

The Compaq Tandem Service Management (TSM) software package consists of server software and client software that provides troubleshooting, maintenance, and services tools for your Compaq *NonStop*™ Himalaya S-series server. TSM client software resides on PC-compatible workstations running the Windows NT 4.0 operating system.

You can also use TSM to perform many of the Guardian tasks described in this guide. For information on using TSM, see the *TSM Online User's Guide.*

Starting the TSM EMS Event Viewer Application

To use a TSM client software component, you must power on the workstation and then start the component you want to use (you can start the TSM client software components in any order). To start the TSM EMS Event Viewer Application:

1. Do one of:
   - On the Windows NT desktop, click `Start` and then choose `Programs>TSM Client>TSM Event Viewer`.
   - In the TSM Low-Level Link Application:
     a. Click System Discovery. TSM displays the Management view window, which shows the physical view of your system.
     b. From the Display menu, choose Events.
   - In the TSM Service Application, from the Display menu, choose Events.

     The TSM EMS Event Viewer Application window appears. (This window is blank until you select search criteria.)
2. From the File menu, choose Log On. TSM displays the Log On to NSK System dialog box.
3. Type a NonStop™ Kernel user name and password and select the NonStopú Himalaya S-series server to which you want to connect and then click OK. When you are connected to the server, the main window status bar displays the message "Connected to Server."
4. From the Setup menu, select the retrieval criteria you want to use to specify which events you want displayed in the window. You can choose to display events during a specific time frame, from specific event logs, or from specific subsystems.

For more information, see the *TSM Online User's Guide* or the TSM EMS Event Viewer Application online help.

Using the Event Viewer

The TSM EMS Event Viewer provides:

- Serial, summary, expanded, and detailed views of events.
- Multiple event sources, including merged and/or imported logs.
- Flexible event selection capabilities (including by subsystem, subject, priority, as well as an exclusion function).
- Flexible event display (single line, full detail, token selection, color/emphasis encoding).
- Filter manipulation functions to refine and focus a view of the events.
- Decoding of events and tokens on the workstation.
- Ability to define and save views.
- Ability to view related events.

You can use the TSM EMS Event Viewer Application to:

- Set Up Event Retrieval Criteria

  The Event Viewer provides dialogs for you to set up the search criteria used to retrieve events based on start and end time, subsystem, source, and specific events. You can save any Event Retrieval Criteria you have specified for future use.

- Select Events for Display

  After retrieving the events from the NonStop™ Kernel system to the workstation, you can further select which events you want to look at. You can:

  - Change event display parameters, including color and font style, size, and type.
  - Display events by timeframe, subsystem, subject, and priority, as well as specify which events should be included in or excluded from the display.
  - Select which columns are to be displayed and specify their order.

  Use these features to further refine the criteria specified for event retrieval.

- View Events

  You can specify how you want to view the selected events. You can view events by log time or in summary form, then select specific events for detailed viewing. You can also display all events related to a selected one.

18

Displaying Version and System Information

This section describes two utilities, VPROC and SYSINFO, which you can use to find information about your system and its software.

VPROC is a utility that uniquely identifies and displays product version information for one or more files. SYSINFO displays basic information about a local or remote system. If you are reporting a problem or asking for product or system support, you need to run VPROC or SYSINFO so you can supply Compaq with the correct system information or product versions for the product files.

This section contains:

Displaying File Version Information

Task 1: Find Product Files

There can be many files for a product in different locations on a system; you want the one that is actively used. Product files are located in both the Guardian and Open System Services (OSS) environments. VPROC can be run from both environments, so the procedure you need to perform depends upon the environment you are using and where the product files are located.

In the Guardian Environment

The following procedure finds the version of a product's object file that you use because of the values in your PMSEARCHLIST TACL variable. Other users might use other versions of the file.

To find the correct object file for a Guardian product, work in the Guardian environment:

1. Start the product from a TACL prompt. Use the NOWAIT run option in combination with the NAME run option so you can start a process but stay in the TACL program.
2. Use the STATUS command to identify the location of the object file that your system referred to when it started the named process.
3. Use the STOP command to stop the process (unless you intend to use it).

The following example shows how, from the TACL program, to find the object file for the version of the File Utility Program (FUP) that you use:

```
1> fup /nowait, name $stein/
2> status $stein

System \FORTY

Process         Pri PFR %WT Userid  Program file            Hometerm
$STEIN    9,145 135 P   000   6,66  $SYSTEM.SYS00.FUP       $ZTN1.#PTY0A9Q
                    Swap File Name: $GERT.#0295
```

In this example, the file your system refers to when starting FUP is $SYSTEM.SYS00.FUP. Now you should stop this FUP process, as shown:

```
3> stop $stein
```

Related product files should also be in the same subvolume as the FUP object file.

Task 2: Select Files for VPROC Processing

VPROC can display information about product files with the following file codes:

File Code	Content
0	Binary files
100	Executable (TNS or TNS/R COFF format) nonnative object files
180	Guardian C data files, or OSS regular files containing text
510	Standard microcode files
700	TNS/R native (COFF or ELF format) relinkable or executable object files
860	TNS/R millicode files
870	TNS/R millicode files
880	TNS/R millicode files

Most of the time you will be interested in the object files for a product. An object file is a program file on disk, such as FUP, TEDIT, or `c89`, that you can run as a process.

Files in the OSS file system can have only the file codes listed previously, so you need not select OSS product files based on their file code. To determine the file code of a file in the Guardian file system, use the FILEINFO command from a TACL prompt, as described in Section 3, Managing Files With TACL.

Files can be stored either singly or in sets called `ar`-format archive files. An `ar`-format archive file is normally used only in the OSS file system and on UNIX systems. If you request version information for an `ar`-format archive file that contains no Binder-format object files, VPROC returns only the timestamp of the most recently modified file in the archive file.

Task 3: Run VPROC

You can run VPROC in the following ways:

- By specifying a full Guardian VPROC command at the TACL prompt in the Guardian environment
- By specifying a full Guardian VPROC command as an operand to the OSS `gtacl` command in the OSS environment
- By specifying a full OSS `vproc` command in the OSS environment
- As an interactive process in either environment

Running VPROC in the Guardian Environment

The following paragraphs describe the Guardian VPROC command when used in the Guardian environment.

For Guardian Files

If you execute VPROC from the TACL program to display version information for Guardian files, the syntax is:

```
VPROC [/run-option/] { file-set | pathname }
```

run-option

Is any run option described under the RUN command in the *TACL Reference Manual*. If you specify *pathname* in the VPROC command, you must enter at least one valid run option. If no run option is needed, use the NAME option.

file-set

Is a file or set of files entered as:

[*system*.[$*volume*.[*subvolume*.]]]*fileid*

The *file-set* parameter specifies the Guardian file name of the product whose version information you are seeking. The *fileid* specification can be either a single disk file ID or an asterisk (*) to indicate all files in a subvolume.

pathname

Specifies the absolute OSS pathname of the product whose version information you seek. A local Guardian file name can be expressed in OSS pathname syntax by replacing the $ with /G/ and replacing the . with /; lowercase characters are used for Guardian file names in OSS pathname syntax (for example, $SYSTEM.SYSTEM.VPROC is expressed as /G/system/system/vproc).

VPROC does not support the wild-card asterisk (*) in OSS pathnames in the Guardian environment.

You cannot express remote Guardian file-set values in OSS pathname syntax.

Note. VPROC retrieves information only for one file-set value or pathname value per command line; additional characters are ignored.

These examples show two ways to display information about FUP in the Guardian environment.

Using Guardian file-name syntax:

```
6> VPROC \MYNODE.$SYSTEM.SYS00.FUP
```

Using OSS pathname syntax:

```
7> VPROC /NAME/ /G/SYSTEM/SYS00/FUP
```

VPROC displays, for example:

```
VPROC - T9617G03 - (30 MAR 1999) SYSTEM \WEST   Date 12 APR 1999, 08:17:34
COPYRIGHT TANDEM COMPUTERS INCORPORATED 1986 -1995


\WEST.$SYSTEM.SYS00.FUP
     Binder timestamp:  23MAR1998 06:20:04
    Version procedure:  S7053D40^29AUG1997^LOAD^10SEP1997
    Version procedure:  T6553D40^28MAR1998^FUP^23MAR1998
           Target CPU:  TNS, TNS/R
      AXCEL timestamp:  23MAR1998 06:20:33
```

The following command requests product version information for a set of Guardian files rather than for a single file. The wild-card asterisk (*) causes VPROC to display information about all object files (file code 100) in the $SYSTEM.SYS00 subvolume.

```
8> VPROC $SYSTEM.SYS00.*
```

For OSS Files

To display version information for local OSS files or for OSS `ar`-format archive files containing Binder-format object files, the syntax is:

```
VPROC / run-option / pathname
```

`run-option`

Is any run option described under the RUN command in the *TACL Reference Manual*. When you specify *`pathname`* in the VPROC command, you must enter at least one valid run option for VPROC to run successfully. If no run option is needed, use the NAME option.

`pathname`

Specifies the absolute OSS pathname of the product whose version information you are seeking. VPROC does not support the wild-card asterisk (*) in OSS pathnames in the Guardian environment.

You cannot express remote OSS pathname values in OSS pathname syntax.

Note. Currently, VPROC has access only to local OSS files.

VPROC can process files in the OSS file system only if VPROC is run on a TNS/R system running a D30.00 or later release with the Open System Services environment active.

This command causes VPROC to display information for an OSS `ar`-format archive file containing Binder-format object files.

```
10> VPROC /NAME/ /USR/LIB/LIBC.B
```

VPROC displays, for example:

```
       Archive member:  versiono
     Binder timestamp:  23SEP1994 16:13:16
    Version procedure:  T8305D30^31OCT1994^OSSUTL^STDLIBS
           Target CPU:  UNSPECIFIED

       Archive member:  rtversno
     Binder timestamp:  23SEP1994 16:13:16
    Version procedure:  T8305D30^31OCT1994^OSSUTL^STDLIBS
           Target CPU:  UNSPECIFIED

       Archive member:  cnlsvprc.o
     Binder timestamp:  23SEP1994 16:13:16
    Version procedure:  T8305D30^31OCT1994^OSSUTL^STDLIBS
           Target CPU:  UNSPECIFIED
```

As stated earlier, a request for version information for an `ar`-format archive file that contains no Binder-format object files returns nothing but the last modified timestamp. For example, if you enter:

```
11> VPROC /NAME/ /USR/LIB/LIBC.C
```

VPROC returns text similar to this:

```
/usr/lib/libc.c
      Last modified timestamp: 10NOV1994 11:47:44
      No VPROC found in this ar-format file
```

Running VPROC in the OSS Environment

VPROC can be run directly in the OSS environment by using the OSS `gtacl` command or indirectly by using the OSS `vproc` command. Both methods are described in the `vproc(1)` reference page either online or in the *Open System Services Shell and Utilities Reference Manual*.

Running VPROC Interactively

VPROC can be run interactively. Starting an interactive VPROC process from the Guardian or OSS environment allows you to retrieve information for multiple, unrelated files.

To start a VPROC process:

- In the Guardian environment, enter VPROC after the TACL prompt.
- In the OSS environment, enter `vproc` after the OSS shell prompt.

VPROC displays text similar to this:

```
VPROC – T9617G03 – (30 MAR 1999)  SYSTEM \NAME     Date 12 April 1999,
COPYRIGHT TANDEM COMPUTERS INCORPORATED 1986 - 1995
Enter filename:
>
```

After the VPROC prompt (>), you can enter a Guardian file-set value or an OSS pathname value for a file in the Guardian or OSS file system. The OSS pathname must be absolute if VPROC is started from the Guardian environment, and VPROC accepts the wild-card asterisk (*) in OSS pathnames only if used from the OSS shell.

To exit VPROC, enter CTRL/Y at the VPROC prompt.

In this example, VPROC displays information about Peruse.

```
12> VPROC
VPROC - T9617G03 - (30 MAR 1999) SYSTEM \WEST   Date 12 AP 1999, 08:17:34
COPYRIGHT TANDEM COMPUTERS INCORPORATED 1986 - 1995

Enter filename:
> $SYSTEM.SYSTEM.PERUSE

$SYSTEM.SYSTEM.PERUSE
     Binder timestamp:  21NOV1995 03:53:08
    Version procedure:  T9101D40^01NOV1995
            Target CPU:  UNSPECIFIED
```

Task 4: Interpret VPROC Output

VPROC returns as much information as it can about a file. Running VPROC can result in either a successful retrieval or a partial or unsuccessful retrieval.

Successful Retrievals

VPROC provides the following product version information in a successful retrieval:

Displayed Label	Displayed Information
`Archive member:`	Indicates the OSS or UNIX archive member file name. This information appears only for `ar`-format archive files.
`Binder timestamp:`	Indicates the time when the object file was generated. This timestamp is either the time when the object was compiled or the time when the object was processed by the Binder program.
`Version procedure:`	Identifies the type of product, product number, and version level. There can be multiple version procedures per product. The version procedure information has this format: *`ctttrvv_ddmmmyy_nnnnnn_xxxxxx`* or *`ctttrvv_ddmmmyyyy_nnnnnn_xxxxxx`* where *`cttt`* is the Tandem number (T number) of the corresponding product. *`rvv`* is the version of the product. *`ddmmmyy`* or *`ddmmmyyyy`* is the release date of the product version. *`nnnnnn`* is the abbreviated product name of the code. *`xxxxxx`* is optional information for use by Compaq.
`Target CPU:`	Indicates the type of system the product runs on, which can be TNS (CISC-based architecture), TNS/R (RISC-based architecture), or UNSPECIFIED.
`AXCEL timestamp:`	Indicates the date and time the code was accelerated. This information appears only for files that have been accelerated.
`GMT Binder timestamp:`	Indicates the date and time the ELF file was created in GMT (UTC).

`Native Mode:`	Indicates whether the file can be executed. This information appears only for files with file code 700. The possible values are:
	`Not runnable file`
	`runnable file`

Partial or Unsuccessful Retrievals

These diagnostic messages appear when VPROC encounters version procedure identification problems.

```
ERROR - NO SUCH VOLUME:  [ filename | pathname ]
```

Cause. Either the Guardian volume that you specified does not exist or you made a typographical error when entering the volume-name portion of the file name or pathname value.

Effect. If you are using VPROC interactively, VPROC prompts you for another file name or pathname. If you are using VPROC from a TACL or an OSS shell prompt, VPROC terminates.

Recovery. Check the volume name and enter a corrected command or value.

```
ERROR:  [ filename | pathname ] does not exist.
```

Cause. Either the specified file does not exist or you made a typographical error when entering the file name or pathname value.

Effect. If you are using VPROC interactively, VPROC prompts you for another file name or pathname. If you are using VPROC from a TACL or an OSS shell prompt, VPROC terminates.

Recovery. Check the file name or pathname and enter a corrected command or value.

```
>>NO T9xxx PROC<<
```

Cause. No version procedure information is stored in the specified file.

Effect. This is an informative message.

Recovery. No action is required.

```
No VPROC found in this ar-format file
```

Cause. No version procedure information is stored in the indicated file.

Effect. This is an informative message.

Recovery. No action is required.

```
Version procedure:  [ filename | pathname ]: Not object file
```

Cause. Either the indicated file does not have a file code of a type that VPROC can read, or the file contains only text data.

Effect. This is an informative message.

Recovery. No action is required.

Displaying System Information

The SYSINFO utility provides basic information about a local or remote system's configuration, including the system name, system serial number, Expand node number, current SYS*nn*, and the software release ID.

Task 1: Run SYSINFO

To display default system information, run SYSINFO by entering the name of the utility at a TACL prompt:

```
7> SYSINFO
```

To display information about another system, run SYSINFO specifying the node name you want information on:

```
8> SYSINFO \node
```

SYSINFO displays, for example:

```
SYSINFO - T9268D37 - (27 NOV 1997) SYSTEM \SNAX Date 12 APR 1999, 08:17:34
COPYRIGHT TANDEM COMPUTERS INCORPORATED 1985, 1987-1997

           System name:  \SNAX
    EXPAND node number:  122
         Current SYSnn:  SYS01
         System number:  D19983
   Software release ID:  D47.00
```

The system serial number is only displayed for a local system, not for a remote system.

Task 2: Interpret SYSINFO Output

SYSINFO provides this information about your system:

Displayed Label	Displayed Information
`System name:`	Indicates the name of the system you requested information on.
`EXPAND node number:`	Indicates the node number on which the system's Expand network connection is running.
`Current SYSnn:`	Indicates the SYS*nn* that the system is currently running on.
`System number:`	Indicates the number of the system you requested information on (available only for the local system).
`Software release ID:`	Indicates the version of the Compaq *NonStop*™ Kernel operating system that is running on the system.

19

Monitoring Hardware Components

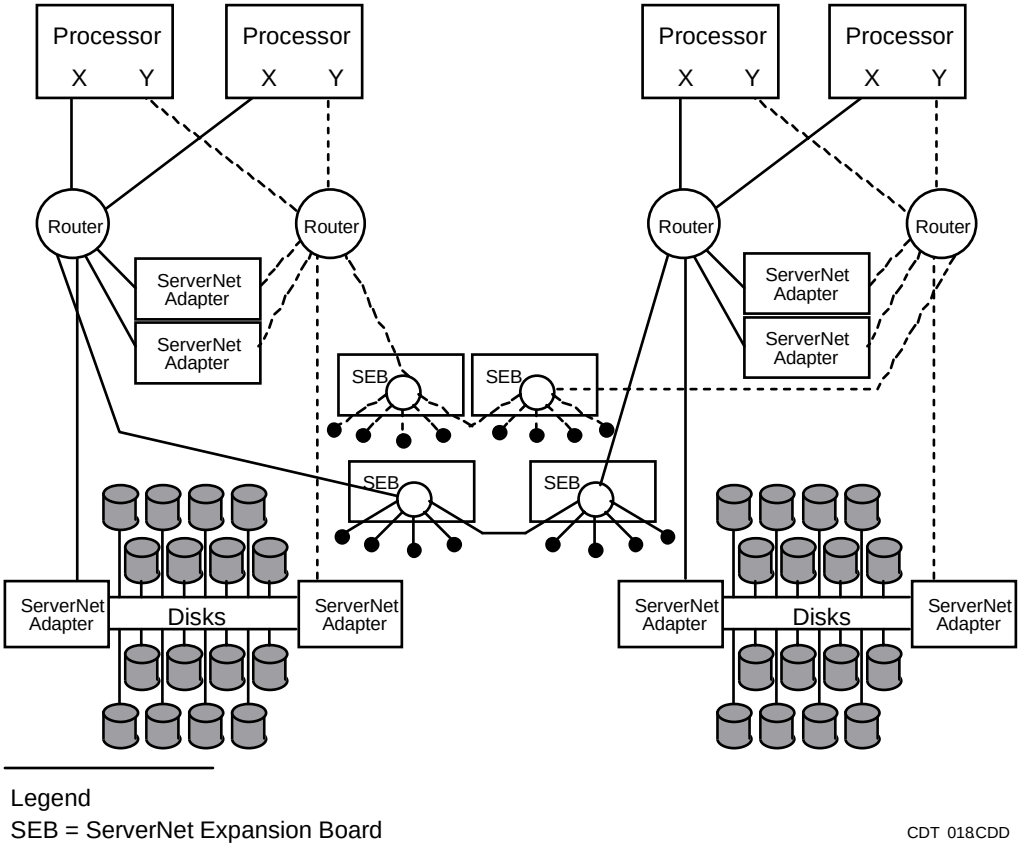
One of your responsibilities as a system operator is to understand how your system is configured so you can check that the hardware and software associated with it are in normal operating condition.

As a standard for what is normal in your operating environment, it is helpful to have copies of:

- System configuration diagrams of your particular system (see the example in Figure 19-1). When you monitor the devices and access paths on your system with the SCF INFO and STATUS commands (explained later in this section), you can use your own system diagram to determine whether all devices and paths are in their correct, or "normal," condition.
- Listings of devices, paths, and processes generated by the SCF INFO and STATUS commands (see the example in Listing the Devices on Your System on page 19-4).
- Listings of devices, paths, and processes generated by the Subsystem Control Facility (SCF).
- A spooler configuration listing (see the example in Section 14, Performing Routine Spooler Operations Using Spoolcom).

This section describes routine system monitoring tasks:

Figure 19-1. Example: Simple System Configuration Diagram



Tools for Monitoring System Status

Many utility programs and tools may be available on your system to assist in your monitoring tasks. Table 19-1 lists system components and the tool or tools you use to monitor each component.

Table 19-1. Tools to Use for System Monitoring

To Monitor:	Use This Tool or Utility:
Devices on your system, including disks, printers, terminals, and tape drives	SCF
Disk free space	Disk Space Analysis Program (DSAP)
Files on your system	File Utility Program (FUP)
Kernel-managed swap files	NSKCOM
Local area networks	SCF
Pathway transaction processing system	PATHCOM
Processes running on your system	TACL PPD and STATUS commands
Processors	ViewSys

Table 19-1. Tools to Use for System Monitoring

To Monitor:	Use This Tool or Utility:
Spooler components, including the supervisor process, collectors, print processes, and so on	SPOOLCOM
Tape drives	MEDIACOM SCF
Terminal communication lines and local area networks	SCF
Transaction Management Facility (TMF)	TMFCOM

Other Useful Tools

Several additional tools might be available for use on your system but are not included in this guide. Examples of such tools include:

- The Event Management Service Analyzer (EMSA) is used to extract specific types of event messages from EMS log files and to create an ENSCRIBE database that you can query to analyze problem trends. For more information, see the *EMS Analyzer User's Guide and Reference Manual*.
- The Measure program is used to collect and display system performance statistics, performance of processors, processes, communication and network lines, files, disks, and terminals. Operations management personnel often use Measure to fine-tune and balance a system. For more information, see the *Measure User's Guide*.
- The PEEK program is use to gather statistical information about processor activity, system storage pools, paging activity, message information, send instructions, and interrupt conditions. For more information, see the *PEEK Reference Manual*.
- The Compaq Object Monitoring Facility (OMF) is used in problem identification and prevention. OMF monitors the same devices as an SCF STATUS command: processors, disks, files, processes, spooler components, audit trails, audit dumps, TMF transactions, and tape mount requests. For more information, see the *Object Monitoring Facility (OMF) Manual*.
- The TSM EMS Event Viewer helps Himalaya S-series users perform many of the tasks associated with viewing and monitoring EMS event logs. It enables you to search for and view the log files in a variety of ways and retrieve events based on start and end time, subsystem, source, and multiple or specific events. You launch the TSM EMS Event Viewer from within the TSM application. For more information, see the *TSM Configuration Guide*.
- The Compaq Tandem Service Management (TSM) client/server application provides troubleshooting, maintenance, and service tools that run on the Himalaya S-series server and on a PC-compatible workstation. TSM combines many of the system maintenance functions provided on D-series releases by the Syshealth toolkit, the Compaq Tandem Maintenance and Diagnostic Subsystem (TMDS), and the Remote Maintenance Interface (RMI) product. For more information, see the *TSM Configuration Guide*.

Listing the Devices on Your System

A device is a physical component of a computer system that is used to communicate with the outside world or to acquire or store data. Examples of devices are printers, disks (volumes), disk drives, tape drives, controllers, processors, and terminals.

Use the Subsystem Control Facility (SCF) to display the configuration characteristics of all the devices on your system. See the *SCF Reference Manual for G-Series Releases* for complete information about SCF, its commands and options, and device types and subtypes.

Use the SCF LISTDEV command to obtain device listings for all device types, subtypes, and other device characteristics. For information about specific devices, such as disks and tapes, use the SCF INFO and STATUS commands.

To list all the devices on your system, enter this command at your TACL prompt:

```
> SCF LISTDEV
```

The format of the LISTDEV display is shown below.

```
LDev   Name      PPID    BPID    Type   RSize  Pri  Program
nnn    $process  nn,nnn  nn,nnn  nn,nn  nnnnn  nnn  \sys.$vol.subvol.file
nnn    $process  nn,nnn  nn,nnn  nn,nn  nnnnn  nnn  \sys.$vol.subvol.file
```

The data shown in the report means:

LDev	The logical device number
Name	The logical device name
PPID	The primary CPU number and process identification number (PIN) of the specified device
BPID	The backup CPU number and process identification number (PIN) of the specified device
Type	The device type and subtype
RSize	The record size the device is configured for
Pri	The priority level of the I/O process
Program	The fully qualified name of the program file for the process

Example

In this example, the SCF LISTDEV command lists all the devices on the system \SHARK.

```
1-> SCF LISTDEV
SCF - T9082F40 - (29FEB96) (01JAN96) - 09/18/96 16:33:11 System \SHARK

Copyright Tandem Computers Incorporated 1986 - 1996

LDev Name      PPID    BPID   Type    RSize Pri Program
   1  $NCP      0,22    0,0    (62,6 )     3 199 \SHARK.$SYSTEM.SYS00.NCPOBJ
   3  $YMIOP    0,6     1,6    ( 6,4 )    80 205 \SHARK.$SYSTEM.SYS00.OSIMAGE
   6  $SYSTEM   0,256   1,256  ( 3,41)  4096 220 \SHARK.$SYSTEM.SYS00.OSIMAGE
  25  $ZZKRN    0,17    1,5535 (66,0 )   132 180 \SHARK.$SYSTEM.SYS00.OZKRN
  26  $ZZWAN    0,288   1,5535 (50,3 )   132 180 \SHARK.$SYSTEM.SYS00.WANMGR
  27  $ZZSTO    0,289   1,5535 (65,96) 64801 180 \SHARK.$SYSTEM.SYS00.TZSTO
  28  $ZZLAN    0,16    1,5535 (43,0 )   132 180 \SHARK.$SYSTEM.SYS00.LANMAN
  30  $ZSNET    0,17    1,5535 (66,0 )   132 180 \SHARK.$SYSTEM.SYS00.OZKRN
  46  $ZM00     0,18    0,0    (45,0 )   132 201 \SHARK.$SYSTEM.SYS00.QIOMON
  48  $TAPE1    0,274   1,5535 ( 4,6 )  2048 200 \SHARK.$SYSTEM.SYS00.OTPPROCP
  49  $TAPE0    0,273   1,5535 ( 4,9 )  2048 200 \SHARK.$SYSTEM.SYS00.OTPPROCP
  52  $DATA12   3,262   2,5535 ( 3,41)  4096 220 \SHARK.$SYSTEM.SYS00.TSYSDP2
  53  $DATA11   3,263   2,5535 ( 3,41)  4096 220 \SHARK.$SYSTEM.SYS00.TSYSDP2
  54  $DATA10   3,264   2,5535 ( 3,34)  4096 220 \SHARK.$SYSTEM.SYS00.TSYSDP2
  60  $DATA04   0,270   1,5535 ( 3,40)  4096 220 \SHARK.$SYSTEM.SYS00.TSYSDP2
  62  $DATA02   0,268   1,5535 ( 3,34)  4096 220 \SHARK.$SYSTEM.SYS00.TSYSDP2
  63  $DATA01   0,267   1,5535 ( 3,40)  4096 220 \SHARK.$SYSTEM.SYS00.TSYSDP2
  64  $DATA00   0,266   1,5535 ( 3,41)  4096 220 \SHARK.$SYSTEM.SYS00.TSYSDP2
  69  $ZZW00    0,295   0,0    (50,0 )     0 199 \SHARK.$SYSTEM.SYS00.CONMGR
  72  $ZZW16    0,296   0,0    (50,4 )   132 180 \SHARK.$SYSTEM.SYS00.WANBOOT
  73  $EXPIP    0,20    1,5535 (63,0 )     3 199 \SHARK.$SYSTEM.SYS00.LHOBJ
  76  $ZNET     0,24    0,0    (50,63)  3900 165 \SHARK.$SYSTEM.SYS00.SCP
  77  $ZTC0     0,297   0,0    (48,0 ) 32000 200 \SHARK.$SYSTEM.SYS00.TCPIP
  78  $ZTNT    0,299   0,0    (46,0 )  6144 170 \SHARK.$SYSTEM.ZTCPIP.TELSERV
```

Determining Device States

This subsection explains how to determine the state of devices on your system and understand the meanings of the various states.

You use the Subsystem Control Facility (SCF) to display the configuration characteristics of all the devices on your system. See the *SCF Reference Manual for G-Series Releases* for complete information about SCF, its commands and options, and device types and subtypes.

To display only the configuration information (attributes and their values) for an object, use the INFO command. INFO does not display dynamic information.

To check the current state of all disk devices on your system, enter:

```
> SCF STATUS DISK $*
```

The format of the STATUS display is shown below.

```
subsystem STATUS object-type object-name

Name          State     PPID    BPID    attr1   attr2   attr3 …
object-name1  state     nn,nnn  nn,nnn  val1    val2    val3  …
object-name2  state     nn,nnn  nn,nnn  val1    val2    val3  …
```

The data shown in the report means:

subsystem	The reporting subsystem name
object-type	The object, or device, type
object-name and Name	The fully qualified name of the object
State	One of the valid object states: ABORTING, DEFINED, DIAGNOSING, SERVICING, STARTED, STARTING, STOPPED, STOPPING, SUSPENDED, SUSPENDING and UNKNOWN.
PPID	The primary CPU number and process identification number (PIN) of the device
BPID	The backup CPU number and process identification number (PIN) of the device
attr	The name of the object attribute
val	The value of the attribute for that object name

Table 19-2 lists and explains the possible object states that can be reported by the SCF STATUS command.

Table 19-2. SCF Object States (page 1 of 2)

State	Substate	Explanation
ABORTING		The object is being aborted. The object is responding to an ABORT command or some type of malfunction. In this state, no new links are allowed and drastic measures may be underway to reach the STOPPED state. This state is irrevocable.
DEFINED		One of the generally defined possible conditions of an object with respect to the management of that object.
DIAGNOSING		The object is in a subsystem-defined test mode entered via the DIAGNOSE command.
INITIALIZED		The system has created the process, but it is not yet in one of the operational states.
SERVICING	SPECIAL	The object is being serviced or used by a privileged process and is inaccessible to user processes.
	TEST	The object is reserved for exclusive testing.
STARTED	(none)	The object is logically accessible to user processes.
STARTING		The object is being initialized and is in transition to the STARTED state.
STOPPED	CONFIG-ERROR	The object is configured improperly.
	DOWN	The object is no longer logically accessible to user processes.

Table 19-2. SCF Object States (page 2 of 2)

State	Substate	Explanation
	HARDDOWN	The object is in the hard-down state or is physically inaccessible because of a hardware error.
	INACCESSIBLE	The object is inaccessible to user processes.
	PREMATURE-TAKEOVER	The backup input/output (I/O) process was asked to take over for the primary I/O process before it had the proper information.
	RESOURCE-UNAVAILABLE	The input/output (I/O) process could not obtain a necessary resource.
	UNKNOWN-REASON	The input/output (I/O) process is down for an unknown reason.
STOPPING		The object is in transition to the STOPPED state. No new links are allowed to or from the object. Existing links are in the process of being deleted.
SUSPENDED		The flow of information to and from the object is restricted (typically, it is prevented). A subsystem must clearly distinguish between the type of information that is allowed to flow in the SUSPENDED state and that which normally flows in the STARTED or STOPPED state. In the SUSPENDED state, the object must complete any outstanding work defined by the subsystem.
SUSPENDING		The object is in transition to the SUSPENDED state. The subsystem must clearly define the nature of the restrictions that this state imposes on its objects.
UNKNOWN		The object's state cannot be determined because the object is inaccessible.

Examples

- Following are some examples of the SCF STATUS command:

```
-> STATUS LINE $LAM3
-> STATUS WS $LAM3.#WS1
-> STATUS WS $LAM3.*
-> STATUS WINDOW $LAM3.#WS1.*
-> STATUS WINDOW $LAM3.*, SEL STOPPED
```

- This example shows the results of the SCF STATUS DISK $* command:

```
29-> STATUS DISK $*
STORAGE - Status DISK \SHARK.$DATA02
LDev   Primary    Backup     Mirror     MirrorBackup   Primary   Backup
                                                       PID       PID
  62   STOPPED    STOPPED    STOPPED    STOPPED         0,268     1,265

STORAGE - Status DISK \SHARK.$DATA03
LDev   Primary    Backup     Mirror     MirrorBackup   Primary   Backup
                                                       PID       PID
  61   STOPPED    STOPPED    STOPPED    STOPPED         0,269     1,264

STORAGE - Status DISK \SHARK.$DATA08
LDev   Primary    Backup     Mirror     MirrorBackup   Primary   Backup
                                                       PID       PID
  56   STOPPED    STOPPED    STOPPED    STOPPED         3,258     2,259

STORAGE - Status DISK \SHARK.$DATA09
LDev   Primary    Backup     Mirror     MirrorBackup   Primary   Backup
                                                       PID       PID
  55   STOPPED    STOPPED    STOPPED    STOPPED         3,265     2,260

STORAGE - Status DISK \SHARK.$DATA10
LDev   Primary    Backup     Mirror     MirrorBackup   Primary   Backup
                                                       PID       PID
  54   STOPPED    STOPPED    STOPPED    STOPPED         3,264     2,261

STORAGE - Status DISK \SHARK.$DATA01
LDev   Primary    Backup     Mirror     MirrorBackup   Primary   Backup
                                                       PID       PID
  63  *STARTED    STARTED   *STARTED    STARTED         0,267     1,266

STORAGE - Status DISK \SHARK.$DATA04
LDev   Primary    Backup     Mirror     MirrorBackup   Primary   Backup
                                                       PID       PID
  60  *STARTED    STARTED   *STARTED    STARTED         0,270     1,263
…
```

Checking the Status of Peripherals

Checking Disk Status

This subsection explains how to list the disk volumes on your system and determine their current status.

You use the Subsystem Control Facility (SCF) to display the configuration characteristics of all the devices on your system. See the *SCF Reference Manual for G-Series Releases* for complete information about SCF, its commands and options, and device types and subtypes.

To list all the disks on your system, enter this command at your TACL prompt:

```
> SCF STATUS DISK $*
```

If you enter the command:

```
> SCF STATUS DISK $DATA00-*
```

you see a report such as the one shown:

```
STORAGE - Status DISK \ALM171.$DATA00-*
LDev  Path          PathStatus  State       SubState     Primary     Backup
                                                           PID         PID
   6 PRIMARY         ACTIVE      STARTED                   0,10        1,10
   6 BACKUP          INACTIVE    STARTED                   0,10        1,10
   6 MIRROR          ACTIVE      STARTING     REVIVE       0,10        1,10
   6 MIRROR-BACKUP   INACTIVE    STARTING     REVIVE        0,10        1,10
```

This report shows that $DATA00:

- Is a mirrored volume (primary and mirror paths)
- Has a mirror disk that is being revived (SubState REVIVE)

The data shown in the report means:

LDev	The logical device number
Path	The disk path assignment
PathStatus	The status of the disk path; whether the disk path is the current path (ACTIVE) or not (INACTIVE)
State	The current SCF state of the disk path
SubState	The current SCF substate of the disk path
Primary PID	The primary CPU number and process identification number (PIN) of the specified device
BPID	The backup CPU number and process identification number (PIN) of the specified device

Other common disk-monitoring operations such as checking disk free space with the Disk Space Analysis Program (DSAP) are included in Section 9, Performing Routine Disk Operations.

Examples: STATUS DISK Command

- To display the summary status of the disk $DATA00, enter:

  ```
  -> STATUS $DATA01
  ```

```
34-> STATUS $DATA01
STORAGE - Status DISK \SHARK.$DATA01
LDev   Primary    Backup     Mirror     MirrorBackup   Primary   Backup
                                                       PID       PID
  63  *STARTED    STARTED   *STARTED    STARTED         0,267     1,266
```

- To display the summary status of the mirror disk of the volume $DATA00, enter:

```
-> STATUS $DATA02-M
```

```
47-> STATUS DISK $DATA02-M
STORAGE - Status DISK \SHARK.$DATA02-M
LDev  Path          PathStatus  State      SubState         Primary   Backup
                                                              PID       PID
  62  MIRROR        INACTIVE    STOPPED    HARDDOWN          0,268     1,265
```

- To display the status of all disks, enter:

```
-> STATUS DISK $*
```

```
29-> STATUS DISK $*
STORAGE - Status DISK \SHARK.$DATA02
LDev   Primary    Backup     Mirror     MirrorBackup   Primary   Backup
                                                       PID       PID
  62   STOPPED    STOPPED    STOPPED    STOPPED         0,268     1,265

STORAGE - Status DISK \SHARK.$DATA03
LDev   Primary    Backup     Mirror     MirrorBackup   Primary   Backup
                                                       PID       PID
  61   STOPPED    STOPPED    STOPPED    STOPPED         0,269     1,264
…
```

- To display the detailed status of the disk $DATA01, enter:

```
-> STATUS $DATA01, DETAIL
```

```
35-> STATUS $DATA01, DETAIL
STORAGE - Detailed Status DISK \SHARK.$DATA01

Disk Path Information:
 LDev  Path          PathStatus  State      SubState         Primary   Backup
                                                              PID       PID
   63  PRIMARY       ACTIVE      STARTED                      0,267     1,266
   63  BACKUP        INACTIVE    STARTED                      0,267     1,266
   63  MIRROR        ACTIVE      STARTED                      0,267     1,266
   63  MIRROR-BACKUP  INACTIVE    STARTED                     0,267     1,266

General Disk Information:
  Device Type........... 3           Device Subtype........... 40
  Primary Drive Type.... 4565-1      Mirror Drive Type........ 4565-1
  Physical Record Size.. 4096        Priority................. 220
  Library File..........
  Program File.......... \SHARK.$SYSTEM.SYS00.TSYSDP2
  Protection............ MIRRORED

Usage Information:
  Capacity (MB)......... 2000.09    Free Space (MB).......... 290.76 (14.53%)
  Free Extents.......... 16          Largest Free Extent (MB). 172.42

Hardware Information:
  Path            Location              Power           Physical Status
                  (group,module,slot)
  PRIMARY         (1,1,3)               DUAL            PRESENT
  MIRROR          (1,1,4)               DUAL            PRESENT
```

Checking Tape Drive Status

This subsection explains how to list the tape drives on your system and determine their status.

Checking Tape Drive Status With MEDIACOM

To check the status of all tape drives on your system with the MEDIACOM program, enter:

```
> MEDIACOM STATUS TAPEDRIVE
```

A listing similar to this example is sent to your home terminal:

```
MEDIACOM - T6028D20 (01JUN93)
Copyright Tandem Computers Incorporated 1993
                 Drive    Tape    Tape     Label  Open
Tape Drive       Status   Name    Status   Type   Mode   Process Name
----------       ------   ----    ------   -----  ----   ------------
$TAPE1           INUSE    TT0046  ASSIGNED ANSI   LP     \SKY.$BURT
$TAPE2           INUSE    TT0047  ASSIGNED ANSI   LP     \SKY.$SID
```

The fields in this display are explained in the *DSM/Tape Catalog Operator Interface (MEDIACOM) Manual*.

Example

To obtain status information about the tape drive $TAPE1 by using MEDIACOM, enter:

```
> MEDIACOM STATUS TAPEDRIVE $TAPE1
```

A listing such as this is sent to your home terminal:

```
MEDIACOM - T6028D20 (01JUN93)
Copyright Tandem Computers Incorporated 1993
                 Drive    Tape    Tape     Label  Open
Tape Drive       Status   Name    Status   Type   Mode   Process Name
----------       ------   ----    ------   -----  ----   ------------
$TAPE1           INUSE    TT0046  ASSIGNED ANSI   LP     \SKY.$BURT
```

This listing shows that $TAPE1 is in use. For more information about MEDIACOM, the listings it generates, and the tasks it enables you to perform, see the *DSM/Tape Catalog Operator Interface (MEDIACOM) Manual* and to Section 10, Using Labeled Tapes.

Checking Tape Drive Status With SCF

To check the status of all tape drives on your system with SCF, enter:

```
> SCF STATUS TAPE $*
```

A listing similar to this is sent to your home terminal:

```
STORAGE - Status TAPE $*
LDev   State      SubState              Primary   Backup    DeviceStatus
                                        PID       PID
  48   STOPPED    DOWN                   0,274
  49   STOPPED    DOWN                   0,273
```

The data shown in the report means:

LDev	The logical device number
State	The current SCF state of the tape path
SubState	The current SCF substate of the tape path
Primary PID	The primary CPU number and process identification number (PIN) of the specified device
Backup PID	The backup CPU number and process identification number (PIN) of the specified device
DeviceStatus	The status of the device path

See Table 19-2 on page 19-6 for more information about the possible states of tape drives and other devices.

For additional information about tape operations and the tasks you can perform, see Section 10, Using Labeled Tapes.

Example

To obtain status information about the tape drive $TAPE1 by using SCF, enter:

```
> SCF STATUS TAPE $TAPE1
```

A listing such as this is sent to your home terminal:

```
STORAGE - Status TAPE $TAPE1
LDev   State      SubState              Primary   Backup    DeviceStatus
                                        PID       PID
  48   STOPPED    DOWN                   0,274
```

Checking Printer and Collector Status

This subsection explains how to list the printers on your system and determine their status. It also explains how to check the status of the spooler subsystem collector processes, which accept output from applications and store the output on a disk.

Checking Printer Status

To check the status of all printers on your system with the SPOOLCOM utility, enter:

```
> SPOOLCOM DEV
```

A listing similar to this is sent to your home terminal:

```
DEVICE                          STATE          FLAGS  PROC    FORM
\SAGE.$S1                       WAITING        H      $SPLX
\SAGE.$S2                       WAITING        H      $SPLX
\AMBER.$S                       WAITING        H      $SPLP
\AMBER.$S2                      WAITING        H      $SPLX
```

“WAITING” in the STATE column indicates that the printer is available to print users’ jobs. See Section 14, Performing Routine Spooler Operations Using Spoolcom, for more information about using Spoolcom.

Checking Collector Process Status

It is important that the collector processes on your spooler subsystem do not become more than about 90 percent full. To check their status, enter:

```
> SPOOLCOM COLLECT
```

A listing similar to this is sent to your home terminal:

```
COLLECT  STATE      FLAGS  CPU   PRI  UNIT  DATA FILE             %FULL
$S       ACTIVE             0 , 1 149  4     $SPOOL.SPL.DATA       40
$S1      ACTIVE             1 , 2 149  10    $SPOOL.SPL.DATA1      28
$S2      ACTIVE             2 , 3 149  8     $SPOOL.SPL.DATA2       0
```

This listing shows that the three collector processes, $S, $S1, and $S2, are active and none is approaching a full state. If the SPOOLCOM COLLECT display shows any collector process approaching 90 percent capacity, jobs must be deleted from the collector in question. See Section 14, Performing Routine Spooler Operations Using Spoolcom, for additional information about this and other tasks relating to the spooler subsystem.

The data shown in the report means:

COLLECT	The collector processes.
STATE	The current state of the collector processes. These states can be ACTIVE, DORMANT, DRAIN, or ERROR.
FLAGS	The current SCF substate of the tape path.
CPU	The CPU number of the collector.
PRI	The execution priority of the collector. The default value is 145.
UNIT	The number of 512-word blocks requested by the collector when it needs more disk space. The default value is 4.
DATA FILE	The name of the disk file where the collector stores jobs.
%FULL	The percentage the disk directory is full.

See Table 19-2 on page 19-6 for more information about the possible states of tape drives and other devices.

Example

To check the status of the printer $LASER with the SPOOLCOM DEV command, enter:

```
> SPOOLCOM DEV $LASER
```

A listing such as this is sent to your home terminal:

```
DEVICE                          STATE          FLAGS  PROC    FORM
$LASER                          WAITING        H      $SPLP
```

This display shows that the printer $LASER is up and available to print users' jobs. See Section 14, Performing Routine Spooler Operations Using Spoolcom, for more information about using Spoolcom.

Checking the Status of Processors

The ViewSys utility provides information about processor activity. Using ViewSys, you can list the processors on your system and determine their status. To use this utility, enter:

```
> VIEWSYS
```

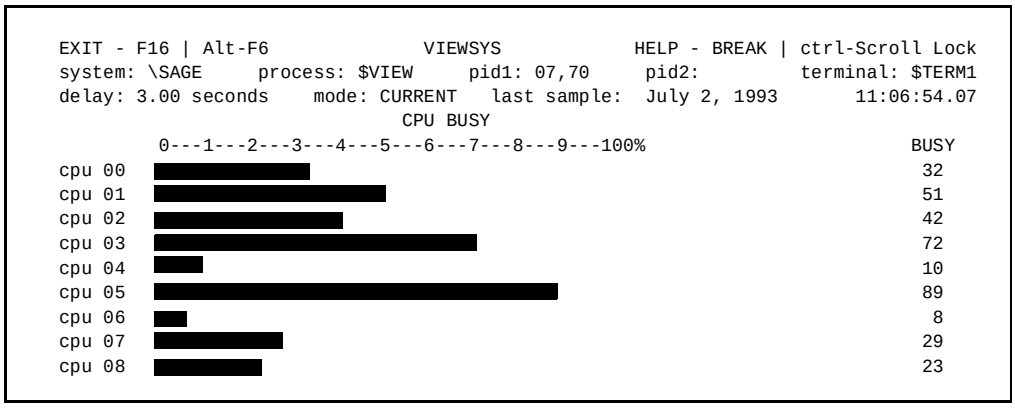
A series of bar graphs that summarize processor performance statistics appears on your terminal.

Note. The Measure utility also collects and displays statistics about system performance and the performance of processors and other system components. Operations management personnel often use this utility to help fine-tune and balance a system. See the *Measure User's Guide* and the *Measure Reference Manual* for instructions on using this utility.

To use the ViewSys utility to obtain information about processor activity, enter:

```
> VIEWSYS
```

After the first ViewSys screen appears, press F1 to view CPU busy statistics:



To exit from the ViewSys utility, press F16.

Checking the Status of Network Components

Checking the Status of Systems in a Network

This subsection explains how to check the status of processors and systems on your network by using the Subsystem Control Facility (SCF). If system users are unable to access another system on your network, you can use the instructions in this subsection to check the status of that system's processors or to check the status of all systems on your network.

- To obtain summary information about all systems and processors on your network, enter:

  ```
  > SCF
  ```

  SCF Banner

  ```
  1-> CPUS
  ```

  A report similar to the following is displayed on your home terminal:

```
 Total      Connected    Total     Up
 Systems    Systems      CPUs      CPUs
 205        196          1150      1071
```

- To check the configuration characteristics, status, and operating-system version of all processors connected to your network, enter:

  ```
  3-> CPUS DETAIL
  ```

  A report similar to the following is displayed on your home terminal:

```
   System       0<--CPU States-->15     OS Version
 000 \AAA       1111,....,....,....        M20
 001 \AAB       1111,11..,....,....        N30
  .    .                .                   .
  .    .                .                   .
  .    .                .                   .
 251 \ABC       ***NOT CONNECTED***
 252 \ABD       1110,....,....,....        M20

 Total      Connected    Total     Up
 Systems    Systems      CPUs      CPUs
 205        196          1150      1071
```

See Interpreting the CPUS Display on page 19-16 for an explanation of the elements in the previous display. See the *SCF Reference Manual for G-Series Releases* for complete information about SCF and its commands.

Interpreting the CPUS Display

The format of the CPUS detailed information display is:

```
    [1]              [2]                    [3]
  System        0<--CPU States-->15     OS Version
116 \C2         1111,....,....,....        P40
```

The numbered elements in this display are:

[1] System contains the numbers and names of the systems known to be connected to the network.

[2] 0<--CPU States-->15 contains symbols indicating the status of each processor in a system:

Symbol	Meaning
1	Processor is active
0	Processor is inactive
Period (.)	Nonexistent processor
NOT CONNECTED	A known system is not currently connected to the network

[3] OS Version contains the version of the operating system running on each system.

For additional information about SCF and its commands, see the *SCF Reference Manual for G-Series Releases*.

Example

To check the configuration characteristics, status, and operating-system version of all processors connected to the system \C2, enter:

```
2-> CPUS \C2
```

A report such as this is displayed on your home terminal:

```
   System        0<--CPU States-->15     OS Version
116 \C2         1111,....,....,....        P40
```

See Interpreting the CPUS Display on page 19-16 for an explanation of the elements in this display.

Checking ServerNet LAN Subsystem Status

The ServerNet LAN Systems Access (SLSA) subsystem supports parallel local-area network (LAN) I/O, allowing S-series servers to communicate across a ServerNet System Area Network (SAN) and access Ethernet devices through various LAN protocols. The SLSA subsystem contains the following SCF objects:

- Processes
- Monitors

- Adapters
- ServerNet Addressable Controllers (SACs)
- Logical Interfaces (LIFs)
- Filters
- Physical Interfaces (PIFs)

The following commands describe how to obtain the status of SACs, adapters, LIFs, and PIFs. For more information on the ServerNet LAN subsystem, see the *LAN Configuration and Management Manual*.

- To check the status of a SAC, enter:

  ```
  > SCF STATUS SAC sac-name
  ```

  A listing similar to this is sent to your home terminal:

```
->STATUS SAC $ZZLAN.E4SA1.0

SLSA Status SAC

Name                Owner   State
$ZZLAN.E4SA1.0        1     STARTED
```

- To check the status of an adapter, enter:

  ```
  > SCF STATUS ADAPTER adapter-name
  ```

  A listing similar to this is sent to your home terminal:

```
->STATUS ADAPTER $ZZLAN.E4SA1

SLSA Status ADAPTER

Name                State
$ZZLAN.E4SA1        STARTED
```

- To check the status of a LIF, enter:

  ```
  > SCF STATUS LIF lif-name
  ```

  A listing similar to this is sent to your home terminal:

```
->STATUS LIF $ZZLAN.LAN0

SLSA Status LIF

Name                State        Access State
$ZZLAN.LAN0         STARTED            UP
```

- To check the status of a PIF, enter:

  ```
  > SCF STATUS PIF pif-name
  ```

A listing similar to this is sent to your home terminal:

```
->STATUS PIF $ZZLAN.E4SA0.0

SLSA Status PIF

Name                State
$ZZLAN.E4SA0.0.A    STARTED
```

Examples

- To obtain a listing of all SACs on $ZZLAN.E4SA1, enter:

  ```
  > SCF STATUS SAC $ZZLAN.E4SA1*
  ```

```
->STATUS SAC $ZZLAN.E4SA1*

SLSA Status SAC

Name                Owner   State
$ZZLAN.E4SA1.0        1     STARTED
$ZZLAN.E4SA1.1        0     STARTED
$ZZLAN.E4SA1.2        0     STARTED
$ZZLAN.E4SA1.3        1     STARTED
```

- To obtain a listing of all adapters on $ZZLAN, enter:

  ```
  > SCF STATUS ADAPTER $ZZLAN.*
  ```

```
->STATUS ADAPTER $ZZLAN.*

SLSA Status ADAPTER

Name                State
$ZZLAN.MIOE0        STARTED
$ZZLAN.E4SA0        STARTED
$ZZLAN.MIOE1        STOPPED
$ZZLAN.E4SA2        STARTED
```

- To obtain a detailed listing of the LIF on $ZZLAN.LAN0, enter:

  ```
  > SCF STATUS LIF $ZZLAN.LAN0 , DETAIL
  ```

```
->STATUS LIF $ZZLAN.LAN0 , DETAIL

SLSA Detailed Status LIF \SYS.$ZZLAN.LAN0

Access State.............. UP
CPUs with Data Path...... ( 0 )
Potential Access CPUs.... ( 0, 1 )
State..................... STARTED
Trace Filename...........
Trace Status.............
```

- To obtain a listing of all PIFs on $ZZLAN.E4SA0, enter:

  ```
  > SCF STATUS PIF $ZZLAN.E4SA0.*
  ```

```
->STATUS PIF $ZZLAN.E4SA0.*

SLSA Status PIF

Name                State
$ZZLAN.E4SA0.0.A    STARTED
$ZZLAN.E4SA0.0.B    STARTED
$ZZLAN.E4SA0.1.A    STOPPED
$ZZLAN.E4SA0.1.B    STARTED
```

Checking ATP6100 Line Status

This subsection summarizes how to check for problems on an ATP6100 line.

The ATP6100 Communications Access Process (CAP) provides the capability to communicate over a variety of asynchronous communication devices (especially terminals and printers) in the 6100 Communications Subsystem (CSS). Application data is passed to one of several protocol modules by ATP6100. The protocol module gives the application extensive control over the line, and each protocol module can control either a single line or multiple lines.

For G-series systems, ATP6100 lines are defined and started with SCF from the WAN subsystem.

You use the SCF interface to the WAN subsystem to:

- Define ATP6100 lines and the modifiers that affect their operation
- Start ATP6100 lines
- Stop, alter, or delete ATP6100 lines
- Monitor the ATP6100 processes and restart them if both the primary and backup processes fail

To check the status of an ATP6100 line, first enter the Subsystem Control Facility (SCF):

```
1> SCF
```

Assume the user's terminal line (the terminal line name is the same as the first component of the user's terminal name, for example, $JT1):

```
-> ASSUME LINE $line
```

Next, assume the subdevice name of the user's terminal (the subdevice name is the same as the second component of the user's terminal name, for example, #J01).

```
-> ASSUME SU #subdevice
```

Check the status of the line and subdevice:

```
-> STATUS LINE
```

For information about the WAN subsystem SCF commands that are relevant to ATP6100 configuration, see the *SCF Reference Manual for Asynchronous Terminals and Printer Processes*.

For information about the functions of the WAN subsystem, see the *LAN Configuration and Management Manual*.

For information on terminal profiles and installing and managing ATP6100 lines, see the *Asynchronous Terminals and Printer Processes Configuration Manual.*

Examples

To check the status of the terminal line and subdevice $JT1.#J01, enter:

```
1> SCF
```

```
1> SCF
SCF - T9082F40 - (29FEB96) (01JAN96) - 09/18/96 16:20:46 System \SHARK
Copyright Tandem Computers Incorporated 1986 - 1996
(Invoking \SHARK.$SYSTEM.NOSUBVOL.SCFCSTM)
```

```
-> ASSUME LINE $JT1
-> ASSUME SU #J01
-> STATUS LINE
```

A status display similar to this appears:

```
ATP6100 STATUS LINE

Name      State       PPID     BPID     I/O Addr       Track Id

$JT1      STARTED    (2,62)   (3,62)    0,5,%3, %0     A00001
```

Checking Line Handler Status

A line handler is a component of your system's data communications subsystem. It is an Expand I/O process that transmits and receives data on a communications line, either directly or by communicating with another I/O process. You can check the status of the line handlers on your system or on another system in your network to which you have remote access.

To check the status of the line handlers on your system, enter:

```
> SCF STATUS LINE $line
```

A listing similar to this example is sent to your home terminal:

```
- > STATUS LINE $LHPLIN1

EXPAND Status LINE
Name            State       PPID       BPID       ConMgr-LDEV
$LHCS6S         STARTED      1, 20      2,25             49
```

Line handlers are commonly named $LH. This listing shows that the Expand line handlers on the system being monitored are up and functioning normally.

The data shown in the report means:

Name	Specifies the name of the PATH object
State	Indicates the summary state of the path. The path is in either the STARTED, STARTING, DIAGNOSING (for T/3880 concentrators only) or STOPPED state.
PPID	Specifies the primary process ID
BPID	Specifies the backup process ID
ConMgr-LDEV	Contains the LDEV of the Concentrator Manager. This field applies only to T/3880 concentrator lines

Check the meaning of the code in Determining Device States on page 19-5 if the State is other than STARTED. Depending upon the type of problem, follow your established procedures for problem reporting and escalation.

For more information about line handlers, see the *Expand Configuration and Management Manual*.

Example

To check the detailed status of line $LHCS6S, enter:

```
-> SCF STATUS LINE $LHCS6S, DETAIL
```

A listing such as this is sent to your home terminal:

```
- > STATUS LINE $LHCS6S, DETAIL

PPID..................... ( 3, 24) BPID................. ( 2, 24)
State.................... STOPPED  Path LDEV...........       50
Trace Status............     OFF  Clip Status......... UNLOADED
ConMgr-LDEV.............      49
Path-prim
Path-alter
```

Checking *NonStop*™ TM/MP Status

This subsection explains how to check the status of the NonStop™ Transaction Manager/MP (TM/MP) and the data volumes it protects.

The main component of NonStop™ TM/MP is the Transaction Management Facility (TMF) subsystem, a standard Compaq tool that maintains the consistency, integrity, and durability of a distributed database that is being updated by concurrent transactions. You use the TMFCOM command interface or the TM View graphical user interface to manage and operate the TMF subsystem, and as a system operator, you might check TMF status in your routine system monitoring.

To monitor the basic components of the TMF subsystem using TMFCOM, enter:

```
> TMFCOM
```

```
~ STATUS TMF
```

A report summarizing the current activity of the TMF subsystem, audit trails, and the audit dump and catalog processes is displayed. See the example below and TMF States on page 19-22 for more information.

The STATUS TMF command presents status information about the audit dump, audit trail, and catalog processes. Thus, in addition to the general TMF information, the STATUS TMF command combines information from the STATUS AUDITDUMP, STATUS AUDITTRAIL, and STATUS BEGINTRANS commands. However, information from the other STATUS commands (STATUS DATAVOLS, STATUS OPERATIONS, STATUS SERVER, and STATUS TRANSACTION) does not appear in the STATUS TMF display.

To display information about the data volumes for which the TMF subsystem generates audit records on behalf of transactions performed on those volumes, enter:

```
~ STATUS DATAVOLS
```

You can control which volumes are displayed, using the STATE, AUDITTRAIL, and RECOVERYMODE parameters.

The normal operating state for a data volume is STARTED, which indicates that the volume is ready to process TMF transactions. Audited requests are allowed for data volumes in this state only as long as transaction processing is enabled within the subsystem.

TMF States

The TMF subsystem can be in any of the states shown in Table 19-3.

Table 19-3. TMF States

State	Meaning
Empty Audit Trail Configuration	The TMF subsystem has been brought up for the first time on this node and thus no configuration exists for it, or a DELETE TMF command was executed.
Configuring New Audit Trails	The TMF subsystem has not yet been started with this configuration.
Stopped	The TMF subsystem is stopped.

Table 19-3. TMF States

State	Meaning
Starting	The TMF subsystem is starting and is in one of the following conditions: ● Services The subsystem is starting audit trail service and other services. ● Waiting for Network Transactions to be Resolved The subsystem is waiting for all network transactions to be resolved. ● Data Volumes The TMF subsystem is starting data volumes. ● Running Backout The subsystem is backing out transactions that must be aborted.
Started	The TMF subsystem has started.
Stopping	The TMF subsystem is stopping and is in one of the following conditions: ● Waiting for Transactions to Finish The subsystem is waiting for all transactions to be completed. ● Data Volumes The subsystem is stopping data volumes. ● Waiting for RDF The subsystem is waiting for the Remote Duplicate Database Facility (RDF) to shut down. ● Services The subsystem is stopping audit trail service and other services.
Deleting	The TMF subsystem is purging its current configuration, audit trails, and volume and file recovery information for the database in response to a DELETE TMF command.

For information about using the graphical user interface, TM View, to monitor your TMF subsystem, see the *NonStop TM/MP: Getting Started with TM View*.

Examples

To check the status of the TMF subsystem on your node, enter:

```
> TMFCOM
```

After TMFCOM displays its banner, enter:

```
~ STATUS TMF
```

TMFCOM responds with a display similar to:

```
TMF Status:
  System: \SAGE, Time: 6-Jul-1994 11:08:06
  State: Started
  Transaction Rate: 0.10 TPS
AuditTrail Status:
  Master:
    Active audit trail capacity used: 55%
    First pinned file: $MAT1.ZTMFAT.AA000044
      Reason: Active transaction(s).
    Current file:  $MAT1.ZTMFAT.AA000045
AuditDump Status:
  Master: State: enabled, Status: active, Process $X545,
    File: $MAT2.ZTMFAT.AA000042
BeginTrans Status: Enabled
Catalog Status:
    Status: Up
```

See TMF States on page 19-22 for an explanation of the elements of this display.

To check the status of all data volumes, enter:

```
~ STATUS DATAVOLS
```

TMFCOM responds with a display similar to:

```
          Audit  Recovery
Volume    Trail    Mode    State
-------------------------------------------------
$DATA1    MAT    Online    Started
$DATA2    MAT    Online    Started
$DATA3    MAT    Online    Recovering
$DATA4    MAT    Archive   Recovering
$DATA5    AUX01  Online    Started
$DATA6    AUX01  Online    Started
$DATA6    AUX01  Archive   Recovering
```

Checking the Status of Pathway

This subsection explains how to check the status of the Pathway transaction processing applications. Pathway is a group of related software tools that enables businesses to develop, install, and manage online transaction processing applications. Several Pathway environments can exist for a system. As a system operator, you might check the status of Pathway in your routine system monitoring. For specific information about Pathway, see the *NonStop TS/MP System Management Manual*.

To learn the names of the Pathway processes running on your system, enter:

```
> STATUS *, PROG $*.*.PATHMON
```

To access PATHCOM to communicate with one of the PATHMON processes, enter:

```
> PATHCOM $pathmon-process-name
```

At the PATHCOM prompt, enter:

```
= STATUS PATHWAY
```

To check the state of the PATHMON process within the Pathway environment, enter:

```
= STATUS PATHMON
```

PATHMON States

The status of the PATHMON process can be either STARTING or RUNNING:

- STARTING indicates that a cold or cool start has not finished.
- RUNNING indicates that a cold or cool start has finished.

  The other elements of the STATUS PATHMON display are as follows:

- CPUS shows the number of the primary and backup processors in which the PATHMON process is running. If the backup PATHMON process is not running, the second number will be blank.
- PATHCTL, LOG1, and LOG2 contain information about the PATHMON control file and the logging files.
- REQNUM contains the PATHMON internal identifiers of application requesters that are currently running in this environment.
- The FILE column identifies the type of requester.
- The WAIT column explains if the process is waiting, which can be caused by one of the following conditions:
  - IO: the request is waiting for an I/O operation to finish.
  - LOCK: the request is waiting for an object that has been locked by another requester.
  - PROG-DONE: the request is waiting for a RUN PROGRAM to finish.

Examples

- To check the status of the PATHMON process for the Pathway environment on your system, enter:

  ```
  > PATHCOM $ZVPT

  $Y290: PATHCOM - T9153D20 - (01JUN93)
  COPYRIGHT TANDEM COMPUTERS INCORPORATED 1980 - 1985, 1987 -
  1992

  = STATUS PATHWAY
  ```

  PATHCOM responds with a display such as:

```
                  RUNNING
EXTERNALTCPS            0
LINKMONS                0
PATHCOMS                1
SPI                     1
                                                    FREEZE
                  RUNNING  STOPPED   THAWED     FROZEN  PENDING
SERVERCLASSES          13        5       18          0        0

                  RUNNING  STOPPED  PENDING
SERVERPROCESSES        13       40        0
TCPS                    1        0        0

                  RUNNING  STOPPED  PENDING  SUSPENDED
TERMS                   1        0        0          0
```

  This display provides information about the number of Pathway processes and servers that are running, stopped, and so forth. For specific information about interpreting this display, see the *NonStop TS/MP System Management Manual*.

- To check the status of the PATHMON process for your application, enter:

  ```
  = STATUS PATHMON
  ```

  PATHCOM responds with a display such as:

```
PATHMON -- STATE=RUNNING   CPUS 6:1
  PATHCTL  (OPEN)     $GROG.VIEWPT.PATHCTL
  LOG1 SE  (OPEN)     $0
  LOG2     (CLOSED)

  REQNUM  FILE     PID              PAID     WAIT
  1       PATHCOM  $Y622            8,001
  2       TCP      $Y898
```

Checking the Size of Database Files

This subsection explains how to check the size of critical database files in order to prevent a “file full” error (error 45) from occurring. For summary information relating to problems you might encounter with files, see Solving Common System Process Problems on page 4-11.

To check the size of any file on your system, enter:

```
> FUP INFO filename, DETAIL
```

A report similar to the following is sent to your home terminal:

```
$DATA.FILES.FILEA                    10 Jul 1993, 14:05
    ENSCRIBE
    TYPE U
    CODE 100
    EXT ( 224 PAGES, 14 PAGES )
    ODDUNSTR
    MAXEXTENTS 370
    BUFFERSIZE 4096
    OWNER 8,255
    SECURITY (RWEP): NUNU, LICENSED
    DATA MODIF:  10 Jul 1994, 14:04
    CREATION DATE:  10 Jan 1994, 14:04
    LAST OPEN:  10 Jul 1994, 14:04
    EOF 267022 (58.2% USED)
    FILE LABEL: 822 (20.2% USED)
    EXTENTS ALLOCATED: 10
```

This report shows that FILEA is 58.2 percent full. When database files become 90 percent full or more, you can modify the file extents dynamically with FUP or perform other procedures as determined by your local system policies.

Example

To check the size of the file DATA1.MEMOS, enter:

```
> FUP INFO DATA1.MEMOS, DETAIL
```

A report such as this is sent to your home terminal:

```
$DATA.DATA1.MEMOS               12 Jul 1994, 14:05
    ENSCRIBE
    TYPE U
    CODE 101
    EXT ( 2 PAGES, 2 PAGES )
    ODDUNSTR
    MAXEXTENTS 16
    BUFFERSIZE 4096
    OWNER 8,255
    SECURITY (RWEP): NUNU
    DATA MODIF:  12 Jul 1994, 14:04
    CREATION DATE:  12 Jan 1994, 14:04
    LAST OPEN:  12 Jul 1994, 14:04
    EOF 567022 (88.2% USED)
    FILE LABEL: 775 (31.6% USED)
    EXTENTS ALLOCATED: 10
```

Note. The allocation of additional extents to any file causes that file to take up more disk space. Before you change the maximum allowable extents for any file, as shown in the following example, check your local procedures to determine whether this is the appropriate action for you to take.

Because this file is nearly 90 percent full, you might want to allocate more extents. To allocate additional extents to the file TRAIL1, enter:

```
> FUP
- ALTER MEMOS, MAXEXTENTS 20
- INFO MEMOS, DETAIL
```

A report such as this is sent to your home terminal:

```
$DATA.DATA1.MEMOS           12 Jul 1993, 14:05
    ENSCRIBE
    TYPE U
    CODE 101
    EXT ( 2 PAGES, 2 PAGES )
    ODDUNSTR
    MAXEXTENTS 20
    BUFFERSIZE 4096
    OWNER 8,255
    SECURITY (RWEP): NUNU
    DATA MODIF:  12 Jul 1993, 14:04
    CREATION DATE:  12 Jan 1993, 14:04
    LAST OPEN:  12 Jul 1993, 14:24
    EOF 567022 (78.5% USED)
    FILE LABEL: 649 (22.8% USED)
    EXTENTS ALLOCATED: 10
```

This report shows that the maximum number of extents allocated to this file has increased to 20 and that the file TRAIL1 is now only 78.5 percent full.

For more information about setting file extents, see the *File Utility Program (FUP) Reference Manual*.

Automating System Monitoring

You can automate many of the procedures described in this section. Automation saves you time and helps you to perform many routine tasks more efficiently.

Your operations environment might be using TACL macros, TACL routines, or command files to perform routine system monitoring and other tasks. These allow you to run many procedures so that you can quickly determine system status, produce reports, or perform other common tasks.

This subsection contains an example of a command file you can use or adapt to check many of the system elements discussed throughout this section. The *TACL Programming Guide* also contains an example that you can use or adapt to automate system monitoring.

To create a command file that will automate system monitoring, enter this into an EDIT file:

```
COMMENT  THIS IS THE FILE SYSCHK
COMMENT  THIS CHECKS ALL DISKS:
SCF STATUS DISK $*
COMMENT  THIS CHECKS ALL TAPE DRIVES:
SCF STATUS TAPE $*
COMMENT  THIS CHECKS THE SPOOLER COLLECTOR PROCESSES:
SPOOLCOM COLLECT
COMMENT   THIS CHECKS THE LINE HANDLERS:
SCF STATUS LINE $*
COMMENT  THIS CHECKS THE STATUS OF TMF:
TMFCOM;STATUS TMF
COMMENT  THIS CHECKS THE STATUS OF PATHWAY:
PATHCOM $ZVPT;STATUS PATHWAY;STATUS PATHMON
```

After you create this file, enter this at your TACL prompt:

```
> OBEY SYSCHK
```

See the example below for an illustration of the display that is sent to your home terminal when you execute this command file.

Example

To execute the command file SYSCHK to automatically monitor many elements of your system discussed in this section, enter:

```
> OBEY SYSCHK
```

Listings similar to this are sent to your home terminal:

```
 COMMENT  THIS IS THE FILE SYSCHK
 COMMENT  THIS CHECKS ALL DISKS:
 SCF STATUS DISK $*

STORAGE - Status DISK \SHARK.$DATA12
LDev   Primary    Backup     Mirror     MirrorBackup   Primary   Backup
                                                       PID       PID
  52  *STARTED    STARTED   *STARTED    STARTED         3,262     2,263

STORAGE - Status DISK \SHARK.$DATA01
LDev   Primary    Backup     Mirror     MirrorBackup   Primary   Backup
                                                       PID       PID
  63  *STARTED    STARTED   *STARTED    STARTED         0,267     1,266

STORAGE - Status DISK \SHARK.$DATA04
LDev   Primary    Backup     Mirror     MirrorBackup   Primary   Backup
                                                       PID       PID
  60  *STARTED    STARTED   *STARTED    STARTED         0,270     1,263

STORAGE - Status DISK \SHARK.$SYSTEM
LDev   Primary    Backup     Mirror     MirrorBackup   Primary   Backup
                                                       PID       PID
   6  *STARTED    STARTED    STOPPED    STOPPED         0,256     1,256
```

```
COMMENT  THIS CHECKS ALL TAPE DRIVES:
SCF STATUS TAPE $*

STORAGE - Status TAPE $TAPE1
LDev   State      SubState              Primary   Backup    DeviceStatus
                                        PID       PID
  48   STARTED                           0,274

STORAGE - Status TAPE $TAPE0
LDev   State      SubState              Primary   Backup    DeviceStatus
                                        PID       PID
  49   STARTED                           0,273

COMMENT  THIS CHECKS THE SPOOLER PRINT DEVICES:
SPOOLCOM DEV

DEVICE                            STATE          FLAGS  PROC    FORM
$LINE1                            WAITING        H      $SPLX
$LINE2                            WAITING        H      $SPLX
$LINE3                            WAITING        H      $SPLX
$LASER                            WAITING        H      $SPLP

COMMENT  THIS CHECKS ALL SACS:
SCF STATUS SAC $*

SLSA Status SAC

Name                  Owner     State
$ZZLAN.E4SA1.0          1       STARTED
$ZZLAN.E4SA1.1          0       STARTED
$ZZLAN.E4SA1.2          0       STARTED
$ZZLAN.E4SA1.3          1       STARTED

COMMENT THIS CHECKS ALL ADAPTERS
SCF STATUS ADAPTER $*

SLSA Status ADAPTER

Name                  State
$ZZLAN.MIOE0          STARTED
$ZZLAN.E4SA0          STARTED
$ZZLAN.MIOE1          STARTED
$ZZLAN.E4SA2          STARTED

COMMENT THIS CHECKS ALL LIFS
SCF STATUS LIF $*

SLSA Status LIF

Name                  State        Access State
$ZZLAN.LAN0           STARTED            UP
$ZZLAN.LAN3           STARTED          DOWN

COMMENT THIS CHECKS ALL PIFS
SCF STATUS PIF $*

SLSA Status PIF

Name                  State
$ZZLAN.E4SA0.0.A      STARTED
$ZZLAN.E4SA0.0.B      STARTED
$ZZLAN.E4SA0.1.A      STOPPED
$ZZLAN.E4SA0.1.B      STARTED
```

```
COMMENT   THIS CHECKS THE LINE HANDLERS:
SCF STATUS LINE $*

COMMENT  THIS CHECKS THE STATUS OF TMF:
TMFCOM;STATUS TMF
TMF Status:
  System: \SAGE, Time: 12-Jul-1994  14:05:00
  State: Started
  Transaction Rate: 0.25 TPS
AuditTrail Status:
  Master:
    Active audit trail capacity used: 68%
    First pinned file: $TMF1.ZTMFAT.AA000044
      Reason: Active transactions(s).
    Current file:  $TMF1.ZTMFAT.AA000045
AuditDump Status:
  Master: State: enabled, Status: active, Process $X545,
    File: $TMF2.ZTMFAT.AA000042
BeginTrans Status: Enabled
Catalog Status:
  Status: Up
Processes Status:
  Dump Files:
  #0:  State: InProgress

COMMENT  THIS CHECKS THE STATUS OF PATHWAY:
PATHCOM $ZVPT;STATUS PATHWAY;STATUS PATHMON

                 RUNNING
EXTERNALTCPS           0
LINKMONS               0
PATHCOMS               1
SPI                    1


                                                      FREEZE
                 RUNNING  STOPPED   THAWED     FROZEN  PENDING
SERVERCLASSES         26        1       27          0        0

                 RUNNING  STOPPED  PENDING
SERVERPROCESSES       39        8        0
TCPS                   6        0        0

                 RUNNING  STOPPED  PENDING  SUSPENDED
TERMS                  4        0        0          0


PATHMON -- STATE=RUNNING   CPUS 4:5
  PATHCTL  (OPEN)     $ZVPT.TRANCNFG.PATHCTL
  LOG1     (OPEN)     $TLOG
  LOG2     (CLOSED)

  REQNUM  FILE     PID              PAID     WAIT
  1       PATHCOM  $Y593            8,001
  2       SPI      $UBIQ            30,1
  3       TCP      $ZTTA
  4       TCP      $ZTTB
  5       TCP      $ZTAA
  6       TCP      $ZTAB
  7       TCP      $ZCMA
  8       TCP      $ZCMB
```

The above listings show that all elements of the system being monitored are up and running normally.

A Problem Solving Techniques

This appendix presents an approach you can use in your operations environment to determine the possible causes of problems, to systematically fix or escalate such problems, and to develop ways of preventing the same problems from recurring. Continuous availability of your Compaq *NonStop*™ Kernel system is important to system users, and your problem-solving processes can help make such availability a reality.

Learning the Cause of a Problem: A Systematic Approach

Determining the cause of a problem on your system is similar to learning about a problem you have with your car. If your car fails to start, what is the first thing you do? You would not replace the engine as your first step. You would start with the easiest, least expensive possibilities, try them, and move to more complex, expensive possibilities only if the easier solutions fail.

With this in mind, the four basic steps in systematic problem solving are:

Tools for Identifying Problems

Several tools might be available on your system to assist you or your operations management in problem identification and tracking, including:

- The Event Management Service Analyzer (EMSA) is used to extract specific types of event messages from EMS log files and to create an ENSCRIBE database that you can query to analyze problem trends. For more information, see the *EMS Analyzer User's Guide and Reference Manual*.
- The Measure program is used to collect and display system performance statistics, performance of processors, processes, communication and network lines, files, disks, and terminals. Operations management personnel often use Measure to help fine-tune and balance a system. For more information, see the *Measure User's Guide* and the *Measure Reference Manual*.
- The PEEK program is used to gather statistical information about processor activity, system storage pools, paging activity, message information, send instructions, and interrupt conditions. For more information, see the *PEEK Reference Manual*.

- The Compaq Object Monitoring Facility (OMF) is used in problem identification and prevention. OMF monitors the same devices as an SCF STATUS command: processors, disks, files, processes, spooler components, audit trails, audit dumps, TMF transactions, and tape mount requests. For more information, see the *Object Monitoring Facility (OMF) Manual*.
- The TSM EMS Event Viewer helps you perform many tasks associated with viewing and monitoring EMS event logs ($0 and $ZLOG). It enables you to search for and view log files in several ways and retrieve events based on start and end time, subsystem, source, and multiple or specific events. The TSM EMS Event Viewer is part of the TSM client/server application, which provides troubleshooting, service, and maintenance, tools that run on a Compaq *NonStop*™ Himalaya S-series server and on a PC-compatible workstation. For more information, see the *TSM Configuration Guide*.

A Problem-Solving Process

Table A-1 shows a worksheet that you can use to help you through the problem-solving process. You can use such a worksheet to:

- Get the facts about a problem
- Find and eliminate the cause of the problem
- Focus on prevention
- Make any appropriate escalation decisions

Copy this worksheet and use it to collect and analyze facts regarding the problem you are experiencing. The results might not tell you exactly what is occurring, but they will narrow down the number of possible causes.

Table A-1. Problem Solving Worksheet

Problem Facts	Possible Causes
What?	
Where?	
When?	
Magnitude?	

Table A-1. Problem Solving Worksheet

Problem Facts	**Possible Causes**
Situation Facts	**Escalation Decision**
Plan to Verify/Fix	
Plan to Prevent and Control Damage	

Task 1: Get the Facts and Log the Problem

The first step in solving any problem is to get the facts. Although it is tempting to speculate about causes, your time is better spent in first understanding the symptoms of the problem.

Task 1.1: Learn the Problem Symptoms

You get the facts by asking questions. Try to get a clear, complete description of problem symptoms by collecting facts about the problem itself. Examples of the questions to ask are:

Category	**Questions to Ask**
What?	What are you having trouble with? What specifically is wrong?
Where?	Where did you first notice the problem? Where has it occurred since you first noticed it? Which applications, components, devices, and people are affected?
When?	When did the problem occur? What is the frequency of the problem? Has this problem occurred before this time?
Magnitude?	Is the problem quantifiable in any way? (That is, can it be measured?) For example, how many people are affected? Is this problem getting worse?

Task 1.2: Learn About the Situation

Collect facts about the situation in which the problem arose. A clear description of the situation that led to the problem could indicate a simple solution. Ask questions such as:

- Who reported the problem and how can this person be contacted?
- How critical is the situation?
- What events led to the problem?
- Has anything changed recently that might have caused the problem?
- What event messages have occurred?
- What is the current configuration of the hardware and software products affected?

Example

This is an example of information you might learn from asking questions like those in Task 1.1:

Question	Answer
What is happening that indicates a problem?	A terminal is hung.
Where is this problem occurring?	In the office of USER.BONNIE. The affected terminal is named $JT1.#C02.
When is this problem occurring?	At 8:30 this morning and also at the same time two days ago. Both times, this occurred after three unsuccessful attempts to log on.
What is the magnitude of this problem?	Intermittent; the problem seemed to disappear on its own when it first occurred two days ago.

Task 2: Find and Eliminate the Cause of the Problem

After you collect your facts and log the problem on your worksheet, you are ready to begin considering the possible causes of a problem.

If you rely on your knowledge and experience, the facts you have collected about the situation, and facts you have collected about the problem, you can begin to list possible causes of the problem.

Task 2.1: Identify the Most Likely Cause

To evaluate the possible causes of any problem, you must compare each with the problem symptoms. Using the problem-solving worksheet gives you a guide for accomplishing this task. In this example:

- Possible causes become column headings.
- Entries you have made in the worksheet's rows indicate whether the cause in question could have produced the problem symptoms you listed on the left.
  - Write "yes" in the appropriate box if a possible cause explains a fact.
  - Write "no" in the appropriate box if a possible cause does not explain a fact.
  - The most likely cause is the one that best explains all the facts, that is, the cause that is answered "yes" in many spaces in the column.

Example

Possible causes of the hung terminal problem in the above example could be:

- A terminal hardware problem
- A stopped or suspended TACL process
- System security, which locks a user out after three unsuccessful logon attempts

This worksheet illustrates further how the problem of a hung terminal can be evaluated and lists some possible causes of such a problem:

Problem Facts	**Possible Causes**		
	Terminal hardware	TACL process	Security
What? Terminal $JT1.#C02 is hung	Yes	Yes	Yes
Where? Office of USER.BONNIE	Yes	Yes	Yes
When? 8:30 today Two days ago at 8:30 After 3 failed logon attempts	 Yes Yes No	 Yes Yes No	 Yes Yes Yes
Magnitude? Intermittent Goes away on its own	 ? ?	 Yes Yes	 Yes Yes

Task 2.2: Verify the Possible Cause

Verify that the most likely cause of this problem is what it seems. In the example, this would indicate a security problem. Do not trust your logic, or the results of the matrix alone: ask yourself what would be the fastest, least expensive, safest, and surest way of verifying the problem.

Task 2.3: Fix the Problem

Once you have determined the most likely cause, try to fix it. Follow through and implement the solution or solutions that are appropriate. If this does not work, continue trying other possible solutions that are reasonable considering time, expense, and safety.

Task 3: Escalate the Problem

If the simple solution you have tried in the previous tasks do not solve the problem, you might consider escalating the problem to get additional help.

Task 3.1: Evaluate the Situation

After you complete each task in the problem-solving process, you must decide whether you can continue by yourself or if you must ask for help. Ask yourself:

- Do I have the authority to resolve this problem?
- Do I have the necessary knowledge?
- Do I have the skill?
- Do I have the time?
- What other people need to become involved, if any?
- Who needs to be informed about the problem’s status?

Task 3.2: Provide Documentation

If you decide to escalate the problem, you might be required to document the problem by providing:

- A problem identification number
- A problem classification
- A complete description and history of the problem
- Diagnostic information such as copies of the event log, results of memory dumps, and so on

You might also have procedures at your site for logging problems. If you have a shift log or problem log, this is the appropriate time for making entries in such a log.

Task 4: Focus on Prevention

Solving problems that occur with your system can be exciting because it is active and stimulating. Preventing problems is often less dramatic. But in the end, prevention is more productive than solving problems. Of course, the more work you do to prevent problems before they arise, the fewer problems will occur at potentially critical times.

These questions provide a framework for your problem-prevention efforts:

- Why did this problem occur? What was the root cause? Were there any contributing causes?
- How serious was the problem?
- What is the likelihood that it will occur again?
- Would it be possible to eliminate the causes of this problem?
- Would it be possible to reduce the likelihood that this problem will occur in the future?
- Could automation tools be used to detect and respond to preliminary symptoms of a problem such as this?
- Can anything be done now to minimize the damage that would result from a reoccurrence of this problem?
- Can the problem resolution process be improved in any way?

Glossary

access path. The route between a processor and a physical device such as a disk, through either port of a dual-ported controller. Each device in your system has one or more access paths configured for it. Devices can have primary and backup access paths that allow nonstop operation of the device.

- The **primary access path** originates in the primary processor and is the route of first choice to a device.
- The **backup access path** originates in the backup processor and is the route of second choice to a device. For single-port devices (devices connected to only one controller), the backup path always becomes the active path in the event of any failure to access the device by the primary path.

Account Quality Planning (AQP) Service. Compaq AQP provides services for improving your current operations management processes.

adapter. See ServerNet adapter.

alias. A definition that can provide an alternate name for a TACL command.

Alliance program. A program that Compaq has developed with third parties to augment Compaq offerings. Alliance partners offer consulting services, products, and application development services.

ANSI format. A standard tape-label format supported on Compaq *NonStop*™ Kernel systems. Other formats include BACKUP and TMF. See also IBM format.

asynchronous. A mode of serial-data transmission in which characters are sent at random and the transmission is not synchronized with a separate clock signal; that is, there is no timing relationship between the end of one character and the start of the next. The data contains extra bits: a start bit to signal the beginning of a byte and one or more stop bits to signal the end of the byte. These start and stop bits allow the receiver to determine the correct synchronization. Contrast with “synchronous.”

ATP6100 Communications Access Process (CAP). ATP6100 Communications Access Process (CAP) provides the capability to communicate over a variety of asynchronous communication devices (especially terminals and printers) in the 6100 Communications Subsystem (CSS). Application data is passed to one of several protocol modules by ATP6100. The protocol module gives the application extensive control over the line, and each protocol module can control either a single line or multiple lines.

ATP6100 subsystem. ATP6100 provides the means for an application program to use asynchronous point-to-point terminals, printers, and other devices connected through the WAN concentrator. A communications access process and download module that provides software for asynchronous point-to-point terminal support.

automatic volume recognition (AVR). The process your system uses to check tape labels when a tape is mounted on a tape drive. AVR checks the label to ensure that the correct tape is mounted on the correct tape drive. If an incorrect tape is mounted, the system rejects the tape and displays an operator message.

AVR. See automatic volume recognition (AVR).

BACKCOPY. A utility program that allows you to duplicate tapes made with the BACKUP utility. With BACKCOPY, you can create up to two duplicate tapes for archiving, distribution, or disaster recovery.

BACKUP. A utility program that copies disk files onto magnetic tape.

backup processor. A processor in a NonStop™ Kernel operating system that communicates with the primary processor, allowing the processors to remain independent. A component failure in one processor has no effect on any other processor. See primary processor.

central processing unit (CPU). See processor.

client/server architecture. A computer architecture that divides work between a client and a server. The client provides application and user interface resources; the server stores, retrieves, and protects data. Client/server architecture enables users to access shared data and resources. Clients and servers run on a local area network. See client/server computing.

client/server computing. A model for distributing applications. Communication takes the form of request and reply pairs, which are initiated by the client and serviced by the server. Client/server computing is often used to connect different types of workstations or personal computers to the host computer system, using supported communications protocols. In the NonStop™ Kernel environment, the Remote Server Call product allows a client process (for example, a workstation application) to access a server (for example, a Pathway server). See client/server architecture.

collector. (1) An EMS collector is an Event Management Service (EMS) process that accepts event messages from subsystems and logs them in an event log. (2) A spooler collector accepts output from applications and store the output on disk. Each spooler must include at least one collector but can contain multiple collectors

command file. An edit file (file code 101) that contains a series of TACL commands in the order you want to execute them. To execute the commands in the file, you either use the OBEY command and give the name of the file, or you name the file as the input file when you run TACL. Using command files is a method of automating operations tasks.

communications subsystem. The combination of data communications hardware and software processes that function together as an integrated unit to provide services and access to wide and local area networks.

compatibility distributor ($Z0) . An EMS distributor process that allows downward compatibility with the operator process that existed previously. The compatibility distributor sends operator messages to a console device during system load.

configuration. (1) The arrangement of cabinets, system components, and peripheral devices into a working unit. (2) The definition or alteration of characteristics of an object.

configuration management. The process of configuring the production system hardware and software to adapt to changes. One of the operations disciplines in the operations management model. See operations management model (OM model).

connection. (1) The path between two protocol modules that provides reliable stream delivery service. (2) For the Compaq Tandem Service Management (TSM) software package, the logical link established between the TSM client software on a workstation and the TSM server software on a NonStop™ Kernel system after a logon sequence has been performed. There are two types of logical connections: service connections and low-level links.

console message. See operator message.

consumer distributor . An EMS distributor process that returns selected event messages to management applications upon request. The consumer distributor is used by applications provided by Compaq and by users to read the EMS log file (or alternate log files) and all or specific event messages, depending on filter specifications loaded to the consumer distributor process. The application can take appropriate action, if necessary, in response to an important event. The TSM EMS Event Viewer application, provided by Compaq, uses the consumer distributor.

controller . (1) A device, sometimes a logic board containing computer chips, consisting of hardware and software that manage a computer function such as disk operations or communications. (2) The access paths from a processor to a controller. On NonStop™ Kernel systems, each controller is dual-ported, meaning that it is connected to two separate processors and that it has two access paths, one to each of the processors to which it is connected.

DCOM. See Disk Compression Program (DCOM).

DEFINE. A named set of attributes and values that allows you to specify information for a process before you start the process.

define process library. A set of TACL routines that allow you to run background server processes so that management applications can send commands to a number of subsystems without the overhead of creating a new server process for each command.

device. A logical or physical entity that can be specifically and uniquely identified and with which a processor can communicate. Examples of devices are printers, disks (volumes), disk drives, tape drives, controllers, processors, and terminals

disk. A physical disk pack mounted on a disk drive. Because they are electromechanical devices, disks are more subject to hardware faults and media errors than other system components and should therefore be monitored frequently. See also volume.

Disk Compression Program (DCOM). A utility program that compresses disk space.

Disk Space Analysis Program (DSAP). A utility program that analyzes displays how space is used on a given disk volume.

Distributed Systems Management (DSM) products. A set of software tools that facilitate management of NonStop™ Kernel systems and Expand networks. These tools include the Distributed Name Service (DNS), the Event Management Service (EMS), the Subsystem Control Facility (SCF) for a variety of subsystems, and the Subsystem Programmatic Interface (SPI).

distributor process. An EMS process that distributes event messages from event logs to requesting management applications, to another collector on this or another node, or to printers, devices, or files. EMS provides the following distributor processes for handling event messages: consumer, forwarding, printing, and compatibility.

DSAP. See Disk Space Analysis Program (DSAP).

DSM. See Distributed Systems Management (DSM) products.

E1SA. See Ethernet 1 ServerNet adapter (E1SA).

EMS. See Event Management Service (EMS).

EMSA. See Event Management Service Analyzer (EMSA).

Error. A utility program that displays the error message associated with a file-system error number.

Ethernet. A local area network that uses the CSMA/CD (carrier sense multiple access with collision detection) access method on a bus topology and is the basis for the IEEE 802.3 standard.

Ethernet 1 ServerNet adapter (E1SA). A ServerNet adapter for Ethernet local area network (LAN) that contains one Ethernet port.

event. A change in some condition in the system or network, whether minor or serious. Events might be operational errors, notifications of limits exceeded, requests for action, and so on.

event log. A file or set of files maintained by EMS to store event messages generated by various subsystems.

Event Management Service (EMS). The processes, procedures, and utilities used to report and log events; to forward, print, and distribute event messages to applications; and to filter, retrieve, and obtain information from event messages.

Event Management Service Analyzer (EMSA). A conversational interface that is used to select and analyze events from EMS log files, such as subsystem ID, event number, text, and start and stop time.

event message. The message generated by a subsystem when a subsystem detects an event that might affect its operation. These messages are generally formatted with tokens.

Expand. Compaq's NonStop™ network that extends the concept of fault-tolerant operation to networks of geographically distributed NonStop™ Kernel systems. If the network is properly designed, communication paths are constantly available, even in the event of a single line or component failure. For G-series systems, the Network Control Process (NCP) and Expand line handler processes are defined and started with SCF from the WAN subsystem.

FAXAdvisor. Compaq FAXAdvisor is a free, automated fax information system that enables you to receive professional services documents, support documents, and product documents by means of a touch-tone telephone and a fax machine.

file identifier. The name of an individual file.

file mode. The mode of operation in which Backup or Restore copy files one at a time.

File Utility Program (FUP). A utility program that allows you to perform a variety of operations on disk files.

forwarding distributor . An EMS distributor process that sends selected event messages to an EMS collector on another network node or, if the node has multiple collectors, to an EMS collector on the same node.

FUP. See File Utility Program (FUP).

GCSC. See Global Customer Support Center (GCSC).

Global Customer Support Center (GCSC). A support organization, formerly the Tandem NonStop Support Center (TNSC), that provides telephone and remote diagnostic support for NonStop™ Kernel customers. There are GCSCs located all over the world.

group manager (n, 255). A user ID that allows a user to control a group of user IDs.

high PIN. A process identification number (PIN) that can range from 256 to an architectural limit of 65534. (PIN 255 is reserved by the system.) See also low PIN.

Himalaya S-series servers. The set of servers in the NonStop™ Himalaya range of servers having product numbers beginning with the letter "S." These servers implement the ServerNet architecture and run the NonStop™ Kernel operating system.

IBM format. A standard tape-label format supported on NonStop™ Kernel systems. Other formats include BACKUP and TMF. See also ANSI format.

input/output (I/O). (1) Data entered into a computer or transmitted out of a computer. (2) The process of entering data into or transmitting data out of a computer.

input/output process (IOP). A running program (part of the operating system) that manages the I/O functions for one or more ServerNet addressable controllers (SACs) of the same type.

input source. The resource from which Subsystem Control Facility (SCF) accepts command input. SCF can accept input from a terminal or a disk file. The initial input source is determined by the form of the RUN command used to initiate SCF. At any time during an SCF session, the input source can be temporarily changed to execute a series of commands from a command file.

International Tandem Users' Group (ITUG). An independent organization of NonStop™ Kernel users that encourages communication and information exchange, establishes a forum for special interest groups, and provides feedback to Compaq regarding users' needs.

internet protocol (IP). A data communications protocol that handles the routing of data through a network, which typically consists of many different subnetworks. IP is connectionless; it routes data from a source address to a destination address. See IP address.

IOP. See input/output process (IOP).

IP. See internet protocol (IP).

IP address. An internet protocol (IP) address. An IP address consists of two parts: a network address, which identifies the network, and a local address, which identifies a host within a network. A network address is concatenated with a host address to form the IP address and uniquely identify a host within a network. IP routes data between source and destination IP addresses.

ITUG. See International Tandem Users' Group (ITUG).

Kernel-Managed Swap Facility (KMSF). A facility for managing virtual memory. Through KMSF, the NonStop™ Kernel opens one or more swap files for each processor and manages the files for all the processes needing them. KMSF receives requests for swap space from the NonStop™ Kernel, and returns swap-space reservations to the Kernel. Processes swap to the kernel-managed swap files as needed. As a process's need for swap space grows, KMSF increases the amount of swap space reserved for the process. When the process no longer needs the space, it is returned to KMSF. See NSKCOM.

KMSF. See Kernel-Managed Swap Facility (KMSF).

labeled tape processing. The general term used for an operations environment that uses labeled tapes, as opposed to unlabeled tapes, for backups of data and other operations tasks and activities.

LAN. See local-area network (LAN).

line. The specific hardware path over which data is transmitted or received. A line can also have a process name associated with it that identifies an input/output process (IOP) or logical device associated with that specific hardware path.

local-area network (LAN). A network that is located in a small geographical area and whose communications technology provides a high-bandwidth, low-cost medium to which low-cost nodes can be connected. One or more LANs can be connected to the system such that the LAN users can access the system as if their workstations were connected directly to it. Contrast with "wide area network (WAN)."

licensed program. A program that has the privileges of the operating system. When a licensed program runs, privileged operations in it can bypass ordinary security interfaces.

LIF. See logical interface (LIF).

logical device. A process used to communicate with a physical device. Logical devices are identified by names and numbers; for example, $DISK1.

logical interface (LIF). A process that allows an application or another process to communicate with data communications hardware.

low PIN. A process identification number (PIN) that can range from 0 through 254. See also high PIN.

macro. A sequence of TACL commands and built-in functions that can contain dummy arguments, thus providing a means for simple argument substitution. When the macro name is given to TACL, TACL substitutes the command sequence for the macro name and replaces any dummy arguments with parameter values supplied to TACL. Macros are used to automate operations tasks.

management application. A program or set of programs that issues commands to subsystems, retrieves event messages, or does both things, to assist in managing a system or a network of systems.

Measure. A performance-measurement tool that lets users collect and examine statistics for a system or network.

MEDIACOM. The operator interface to the Distributed Systems Management/Tape Catalog (DSM/TC) product and to the STK 4400 Automated Cartridge System (ACS). It can be used in environments other than DSM/TC. Refer to the *DSM/Tape Catalog Operator Interface (MEDIACOM) Manual* for complete documentation on this utility.

mirrored volume. See volume.

network. Two or more computer systems (nodes) connected so that they can exchange information and share resources.

NLCHECK. A state in labeled-tape processing. If NLCHECK is on, you must give permission for an unlabeled tape operation to open a tape drive by accepting the mount request through MEDIACOM. If NLCHECK is off, the unlabeled tape open can finish without permission. Regardless of the setting, you can perform labeled-tape operations.

NO UNLOAD. A state in labeled-tape processing. When you set NOUNLOAD, tapes remain online after labeling operations. When you clear NOUNLOAD, the system unloads tapes after it completes labeling operations.

node. A physical computer system that is part of an Expand network. It can be a stand-alone system or be part of a network of systems.

nonsensitive command. A command that can be issued by any user or program that is allowed access to the subsystem—that is, a command on which the subsystem imposes no further security restrictions. For Subsystem Control Facility (SCF) subsystems, nonsensitive commands are those that cannot change the state or configuration of objects; most of them are information commands. Contrast with “sensitive command.”

NonStop™ TM/MP. See NonStop™ Transaction Manager/MP (NonStop™ TM/MP).

NonStop™ Transaction Manager/MP (NonStop™ TM/MP). A database-protection subsystem incorporated into the operating system. NonStop™ TM/MP maintains the consistency and integrity of a distributed database that is updated by concurrent transactions.

NonStop™ Virtual Hometerm Subsystem (VHS). A subsystem that acts as a virtual home terminal for applications by emulating a 6530 terminal. NonStop™ VHS receives messages normally sent to the home terminal, such as displays and application prompts, and uses these messages to generate event messages for EMS, which can in turn be used to inform operations staff of problems.

NSKCOM. The command interface to the Kernel-Managed Swap Facility (KMSF). NSKCOM is the primary tool for monitoring, configuring, and managing kernel-managed swap files. See Kernel-Managed Swap Facility (KMSF).

NSX. See Tandem Network Statistics Extended (NSX).

OBEY file. See command file.

object. (1) One or more of the devices, lines, processes, and files in a NonStop™ Kernel subsystem; any entity subject to independent reference or control by one or more subsystems. (2) In Subsystem Control Facility (SCF), a resource controlled by an SCF subsystem. SCF objects include processes, disks, disk files, and data communications lines. Each object has an object type and an object name.

object type. The category of Subsystem Control Facility (SCF) objects to which a specific SCF object belongs; for example, a specific disk has the object type DISK and a specific terminal may have the object type SU. An SCF subsystem has a set of object types for the objects it manages.

offline. Used to describe tasks that can be performed only when the system is down. Contrast with online.

OM model. See operations management model (OM model).

OMF. See Tandem Object Monitoring Facility (OMF).

online. Used to describe tasks that can be performed while the system is up. Contrast with offline.

online-ready site. A fully operational backup site (also known as a hot site) that has all necessary hardware and software. Archived data is sent to the operational-ready site but is not loaded onto the system until a disaster occurs.

ONS. See Open Notification Service (ONS).

Open Notification Service (ONS). (D-series only) A data encapsulation and forwarding server that gathers EMS events from the system event log, translates them into Simple Network Management Protocol (SNMP) trap format, and forwards them to the SNMP Agent, thereby facilitating delivery of NonStop™ Kernel subsystem-specific data to problem management components that comply with SNMP and that are external to the NonStop™ Kernel system.

Open System Services (OSS). An open system environment available for interactive or programmatic use with the NonStop™ Kernel. Processes that run in the OSS environment use the OSS application program interface; interactive users of the OSS environment use the OSS shell for their command interpreter.

operational-ready site. A fully operational backup site (also known as a hot site) that has all necessary hardware and software. Archived data is sent to the operational-ready site but is not loaded onto the system until a disaster occurs.

operations area. The area where you locate the computer systems and peripherals, for example a computer room or an office.

operations management. The operation and management of systems and networks in support of your business. Planning for operations management includes establishing and fulfilling service-level agreements, defining and understanding the OM model, and optimizing the features of NonStop™ Kernel systems and software. See operations management model (OM model).

operations management activities. Activities, as defined by the NonStop™ Kernel operations management model, that support a production system, plan for all aspects of the production system, control the introduction of change into the production system, and operate the production system.

operations management model (OM model). A model for managing NonStop™ Kernel systems that categorizes operations management functions into the following disciplines: production management, problem management, change management, configuration management, security management, and performance management.

operator message. The text displayed for a system operator that describes an event.

operations outage class. An outage class that includes errors caused by operations personnel due to accidents, inexperience, or malice. See outage class.

OSS. See Open System Services (OSS).

outage. Time during which the system is not capable of doing useful work because of a planned or unplanned interruption. From the end-user's perspective, an outage is any time the application is not available.

outage class. A concept developed by Compaq to categorize the cause of unplanned and planned outages. There are five outage classes: physical, design, operations, environmental, and reconfiguration.

outage log. A record of system outages. An outage log can provide an accurate assessment of availability. Compaq recommends that outages be measured in minutes rather than percentages.

outage minutes. A metric recommended by Compaq for measuring outages. Translates percentages into minutes of down time per year.

output destination. The resource to which Subsystem Control Facility (SCF) sends its responses to commands. SCF can direct output to a disk file, an application process, a terminal, or a printer. The initial output destination is determined by the form of the RUN command used to initiate SCF. The output destination can be changed dynamically during an SCF session.

path. The route between a processor and a subsystem. If a subsystem is configured for fault tolerance, it has a primary path (from the primary processor) and a backup path (from the backup processor).

PATHCOM. The interactive administrative interface to the NonStop™ Transaction Services/MP core service.

PATHMON process. The central control process for the NonStop™ TS/MP transaction-processing core service and the optional Pathway/TS software, which together form the Pathway environment. The PATHMON process controls all processes and devices in the Pathway environment and provides the means to configure, manage, monitor, and change the configuration of the Pathway environment.

Pathway environment. The programs and operating environment required for developing and running online transaction-processing (OLTP) applications. This group of tools is packaged as two separate products: NonStop™ TS/MP and the optional Pathway/TS software.

Pathway Open Environment Toolkit (POET). A set of programs and utilities that assist in the creation and running of client/server applications for NonStop™ Kernel systems.

Pathway/TS. A Compaq product that provides tools for developing and interpreting screen programs to support OLTP applications in the Guardian operating environment.

PEEK. A utility program that reports statistical information concerning processor activity for system storage pools, paging activity, send instructions, and interrupt conditions.

Peruse. A spooler utility that allows users to control and monitor print jobs.

performance management. Activities that manage the performance of the production system and network environment to ensure that the systems meet the business needs defined by operations service-level agreements. One of the operations disciplines in the operations management model. See operations management model (OM model).

peripheral device. Any device used by processors to communicate with users or to acquire or store data. Peripheral devices include terminals, printers, workstations, disk drives, and tape drives.

physical interface (PIF). The hardware components that connect a system node to a network.

physical outage class. An outage class that includes physical faults or failure in the hardware. Any type of hardware-component failure belongs in this category. See outage class.

PIF. See physical interface (PIF).

PIN. See performance management..

planned outage. Time during which the system is not capable of doing useful work because of a planned interruption. A planned outage can be time when the system is brought down to allow for servicing, upgrades, backup, or general maintenance. Contrast with unplanned outage.

POET. See Pathway Open Environment Toolkit (POET).

primary processor. The processor that is designated at system generation time as “owning” the controller connected to the two separate processors of a NonStop™ Kernel operating system. The primary processor is the processor that has direct control over the controller. See backup processor.

printing distributor. An EMS distributor process that sends operator messages to printers, devices, processes, or files.

problem management. Activities that provide support for resolving problems in a production environment. One of the operations disciplines in the operations management model. See operations management model (OM model).

process identification number (PIN). An integer that identifies a process in a particular processor and can range from 0 to an architectural limit of 65534.

processor. (1) The central processing unit (CPU). The processor reads program instructions, moves data between processor memory and the ServerNet addressable controllers (SACs), and performs arithmetic operations. (2) One or more computer chips, typically mounted on a logic board, that are designed to perform data processing or to manage a particular aspect of computer operations. Most NonStop™ Kernel systems have a minimum of two processors that can act as backups to each other.

production management. The set of regularly scheduled activities that keeps the applications on a system or network of systems running smoothly. These activities include administering storage media such as disks and tapes, managing space in processors and disks, and starting or stopping system components. One of the operations disciplines in the operations management model. See operations management model (OM model).

reconfiguration. See reconfiguration outage class.

reconfiguration outage class. An outage class that includes all planned outages. Examples include down time required for planned maintenance such as software upgrades, and configuration changes such as adding a new disk or restructuring a database. See outage class.

remote mirroring. A pair of mirrored disk drives that are used together as a single logical drive in which the primary drive and the backup (mirror) drive are located in geographically distinct (remote) locations. Each byte of data written to the primary drive is also written to the mirror drive. If the primary drive fails, the mirror drive can continue operations. By providing geographic separation of mirrored volumes, remote mirroring protects the database from local environment hazards.

SAC. See ServerNet addressable controller (SAC).

SAN. See system area network (SAN).

Safeguard. A group of programs that supplements the security features of the system, providing users of NonStop™ Kernel systems and distributed networks with a set of services for protecting the components of the system or network from unauthorized use. Safeguard services include authentication, authorization, and auditing.

SCF. See Subsystem Control Facility (SCF).

SCP. See Subsystem Control Point (SCP).

scratch tape. A labeled tape whose expiration date has passed. Scratch tapes are used for output.

security management. Activities that provide support for establishing and maintaining system security. One of the operations disciplines in the operations management model. See operations management model (OM model).

sensitive command. A Subsystem Control Facility (SCF) command that can be issued only by a user with super-group access, by the owner of the subsystem, or by a member of the

group of the owner of the subsystem. The owner of a subsystem is the user who started that subsystem (or any user whose application ID is the same as the server ID— the result of a PROGID option that requires super-group access). Contrast with nonsensitive command.

server. (1) An implementation of a system used as a stand-alone system or as a node in an Expand network. (2) A combination of hardware and software designed to provide services in response to requests received from clients across a network. For example, the NonStop™ Himalaya range of servers provides transaction processing, database access, and other services. (3) A process or program that provides services to a client or a requester. Servers are designed to receive request messages from clients or requesters; perform the desired operations, such as database inquiries or updates, security verifications, numerical calculations, or data routing to other computer systems; and return reply messages to the clients or requesters. A server process is a running instance of a server program.

ServerNet adapter. A customer-replaceable unit (CRU) that connects peripheral devices to the rest of the system through a ServerNet bus interface (SBI). A ServerNet adapter is similar in function to an I/O controller logic board (LB) and backplane interconnect card (BIC) in NonStop™ Himalaya K-series servers.

ServerNet address. (1) An algorithmic translation of the memory address indicating where the memory access needed by a ServerNet transaction begins. This is the address contained in ServerNet packets. (2) An identifier for an endpoint on the ServerNet system area network (ServerNet SAN). This address consists of a ServerNet node ID and an identifier that is unique within that ServerNet node.

ServerNet addressable controller (SAC). A controller that is uniquely addressable within one or more ServerNet address domains (SADs) through the node ID and address fields in a request packet. A SAC is typically implemented on some portion of a processor multifunction (PMF) customer-replaceable unit (CRU), an I/O multifunction (IOMF) CRU, or a ServerNet adapter.

ServerNet LAN systems access (SLSA) subsystem. The software that allows the protocol I/O processes (IOPs) and drivers to access the ServerNet adapters.

ServerNet System Area Network (SAN). A wormhole-routed, full-duplex, packet-switched, point-to-point network designed with special attention to reducing latency and ensuring reliability. The ServerNet SAN provides the communication path used for interprocessor messages and for communication between processors and I/O devices.

ServerNet Wide Area Network (SWAN) concentrator. A Compaq data communications peripheral that provides connectivity to a NonStop™ Himalaya S-series server. The SWAN concentrator supports both synchronous and asynchronous data over RS-232, RS-449, X.21, and V.35 electrical and physical interfaces.

service connection. A connection between the Compaq Tandem Service Management (TSM) client software running on a workstation and the TSM server software on a NonStop™ Himalaya S-series server. A service connection can be used only to communicate with the server when the NonStop™ Kernel operating system is running. A service connection provides a comprehensive service and maintenance picture of the server and is used to perform most service management tasks.

service-level agreements. Agreements between the operations group and the group's users that specify the group's objectives, requirements, and standards.

Simple Network Management Protocol (SNMP). An asynchronous request/response protocol (implemented in the Transmission Control Protocol/Internet Protocol (TCP/IP) protocol suite) used for network management. In the SNMP management framework, each managed node is viewed as having several variables. By reading these variables, the managed node is monitored. By changing the value of these variables, the managed node is controlled.

SLSA subsystem. See ServerNet LAN systems access (SLSA) subsystem.

SNAX/HLS. A tool that provides a general-purpose, high-level interface by which NonStop™ Kernel application programs can communicate with intelligent SNA devices and software products.

SNMP. See Simple Network Management Protocol (SNMP).

SPOOLCOM. A spooler utility program that helps system operators monitor and maintain the spooler and create and initialize spooler components.

spooler. A set of programs that acts as an interface between users (and user application) and the print devices on a particular node.

spooler supervisor. A process that monitors and communicates with the other processes and decides when and where to print jobs. Each spooler has one supervisor.

state. In Subsystem Control Facility (SCF) subsystems, one of the generally defined possible conditions of an object with respect to the management of that object. Examples of states are DEFINED, STARTED, and STOPPED.

structured file. A file designed to contain a database. Structured files are key-sequenced, entry-sequenced, or relative files.

substate. Further information about the state of a device. The state and substate together provide information about the current condition of a device or path to a device.

subsystem. (1) A secondary or subordinate system, usually capable of operating independently of or asynchronously with a controlling system. (2) A program or set of processes that manages a cohesive set of Subsystem Control Facility (SCF) objects. Each subsystem has a manager through which applications can request services by issuing commands defined by that subsystem.

Subsystem Control Facility (SCF). A utility program used to control a variety of subsystems. For G-series systems, you can use SCF online to configure, control, and display information about configured objects within SCF subsystems. SCF has been enhanced to perform many of the functions performed on D-series systems by DSC/COUP and PUP.

Subsystem Control Point (SCP). The management process for all NonStop™ Kernel data communications subsystems. There can be several instances of this process. Applications using the Subsystem Programmatic Interface (SPI) send all commands for data communications subsystems to an instance of this process, which in turn sends the commands on to the manager processes of the target subsystems. SCP also processes a few commands itself. It provides security features, version compatibility, support for tracing, and support for applications implemented as NonStop™ process pairs.

subvolume. A group of related files stored on a single disk volume; all the files have the same volume and subvolume name, but each file has a unique file identifier.

subvolume defaulting. The practice of making the system provide the subvolume identifier in a file name by providing only the volume name followed by the file identifier. A D-series file-system interface interprets *volume-name.file-ID* as *volume-name.subvolume-name* with the *file-ID* missing.

super-group user (255, *n*). A user ID that allows users to execute some potentially destructive commands. The super-group user is provided for operators who perform system operations tasks, such as controlling the status of peripherals and other system components.

super ID (255,255). A user ID that allows users to do anything on the system. Users with the super ID can access all data and devices.

support area. The area where the operations staff is located.

SWAN. See ServerNet Wide Area Network (SWAN) concentrator.

swap space. Each processor has at least one kernel-managed swap file that provides the swap space needed by its processes. Proper configuration and management of kernel-managed swap space is critical to the operation of your system.

system. A node. All the processors, controllers, firmware, peripheral devices, software, and related components that are directly connected together to form an entity that is managed by one operating system image and operated as one computer.

system area network (SAN). A high-speed network, within a system, that connects processors to each other and to peripheral controllers. A SAN has the performance of a massively parallel interconnect, but has distribution capabilities similar to a local area network (LAN). NonStop™ Himalaya S-series servers support the ServerNet system area network (ServerNet SAN).

TACL. See Tandem Advanced Command Language (TACL).

Tandem Advanced Command Language (TACL). A powerful, extended command interpreter for the Guardian environment that enables users to perform work on NonStop™ Kernel systems, such as defining aliases, macros, and function keys. TACL also functions as a programming language.

Tandem Capacity Model (TCM). A Compaq product that provides computer-assisted capacity planning. It uses the Microsoft Excel spreadsheet with Measure data to explore different growth scenarios and system configurations.

Tandem Network Statistics Extended (NSX). A network management tool that provides operators with a global perspective on the entire network. With NSX, operators can collect and monitor up-to-the-moment performance statistics on all nodes, processors, and Expand line handlers in the network.

Tandem NonStop™ Kernel. The operating system for NonStop™ systems, which consists of the core and system services. The operating system does not include any application program interface.

Tandem NonStop™ Series (TNS). NonStop™ Kernel computers that support the Guardian environment and that are based on complex instruction-set computing (CISC) technology. The term TNS can refer to the instruction set, the architecture, or the processors. Systems with these processors are the NonStop™ II, NonStop™ TXP, NonStop™ VLX, NonStop™ Cyclone, and NonStop™ CLX 600, CLX 700, and CLX 800 systems. Contrast with Tandem NonStop™ Series/RISC (TNS/R).

Tandem NonStop™ Series/RISC (TNS/R). NonStop™ Kernel computers that support the Guardian environment and that are based on reduced instruction-set computing (RISC) technology. TNS/R processors implement the TNS/R instruction set and maintain architectural compatibility with TNS processors. The term TNS/R can refer to the instruction set, the architecture, or the processors. The first TNS/R processor is the NSR-L processor. Systems with these processors are the NonStop™ Cyclone/R and NonStop™ CLX 2000 systems. Contrast with Tandem NonStop™ Series (TNS).

Tandem NonStop™ Support Center (TNSC). See Global Customer Support Center (GCSC).

Tandem Object Monitoring Facility (OMF). A Compaq product that enables operators to supervise objects such as processors, disks, files, and processes within the NonStop™ Kernel environment.

Tandem Service Management (TSM). A client/server application that provides troubleshooting, maintenance, and service tools for the NonStop™ Himalaya S-series server. Compaq Tandem Service Management (TSM) consists of software components that run on the NonStop™ Himalaya S-series server and on a PC-compatible workstation.

Tandem TCP/IP subsystem. Tandem TCP/IP lets you use the NonStop™ Kernel host from Macintosh, personal computer, and UNIX workstations. Cooperative applications can partition function so as to use the strengths of the different operating systems. Applications running on a NonStop™ Kernel system or an Expand network can transparently exchange data with TCP/IP devices.

Tape label. A record at the beginning of a tape that identifies the tape volume and the files it contains.

Tape processing modes. One of the following three modes that you can use to run jobs in a labeled-tape environment:

- Standard label processing, or **LP mode**: an application generates a mount request at the operator console for a specific labeled tape. You can either mount the requested labeled tape or reject the request.
- Bypass label processing, or **BLP mode**: an application generates a request at the operator console for a specific tape drive. You can accept or reject the request. If you accept the request, the tape mounted on the drive is used for the application. The system does not check the tape, so you must ensure that the correct tape is mounted.
- Unlabeled or no label processing, or **NL mode**: for unlabeled tape processing only, an application generates a drive-usage request at the operator console for an unlabeled tape on a specific tape drive. (Labeled tapes are rejected.)

TCM. See Tandem Capacity Model (TCM).

TCP/IP. TCP/IP is the name applied to the combined protocol layers that are defined by RFC 793 and 791. See Tandem TCP/IP subsystem.

TERM. A task that uses a screen program to control input and output devices (such as terminals or workstations) or input and output processes (such as front-end processes). Each task runs as a thread in a terminal control process (TCP), which can handle many such tasks concurrently.

Terminal control process (TCP). A multithreaded process supplied with Pathway/TS that interprets and executes screen program instructions for each input-output (I/O) device or process the TCP is configured to handle. The TCP coordinates communication between screen programs and their I/O devices or processes and, with the help of the PATHMON process, establishes links between screen programs and Pathway server processes.

TIM. See Total Information Manager.

TMF. See TNS/R.

TMFCOM. The NonStop™ TM/MP command interpreter.

TNS. See Tandem NonStop™ Series (TNS).

TNSC. See Global Customer Support Center (GCSC).

TNS/R. See Tandem NonStop™ Series/RISC (TNS/R).

Total Information Manager. The Total Information Manager (TIM) product integrates multiple collections of NonStop™ Kernel product and support information—including customer manuals, education course information, and other technical documents—to provide a single, searchable library. The TIM viewer provides the interface to collections of documents that are available on local CD-ROM discs as well as online, Internet-accessible servers. This common interface allows you to merge local and online searches and display local and online windows.

Transaction Management Facility (TMF). A standard NonStop™ Kernel tool that maintains the consistency, integrity, and durability of a distributed database that is being updated by concurrent transactions.

TSM. See Tandem Service Management (TSM).

TSM application. A component of the Compaq Tandem Service Management (TSM) client software. The TSM application enables you to communicate with a NonStop™ Himalaya S-series server even when the NonStop™ Kernel operating system is not running. When the operating system is running, you will usually communicate with the server using a service connection. When the operating system is not running, communication must take place over a low-level link.

TSM application notification. A type of notification. Compaq Tandem Service Management (TSM) application notifications are generated by the TSM server software on a NonStop™ Himalaya S-series server when something occurs that might affect the performance of a resource managed by TSM. The TSM Notification Director passes TSM application notifications to the TSM Notification Director where they are used to show up-to-date resource information.

TSM client software. The portion of the Compaq Tandem Service Management (TSM) package that resides on a workstation. The TSM client software consists of the TSM application, the TSM Notification Director, and the TSM EMS Event Viewer. See TSM server software.

TSM EMS Event Viewer. Used to perform a variety of tasks associated with viewing and monitoring EMS event logs. The TSM EMS Event Viewer lets you select from a variety of parameters to set the criteria to search for and view the EMS event log file.

TSM Notification Director. A component of the Compaq Tandem Service Management (TSM) client software. The TSM Notification Director receives notifications and incident reports from a NonStop™ Himalaya S-series server, displays them, and allows you to take action or forward the incident reports to your service provider for resolution. The TSM Notification Director runs on the TSM workstation at all times, even when TSM is not being used.

TSM server software. The major component of the Compaq Tandem Service Management (TSM) package that runs on a NonStop™ Himalaya S-series server. When the NonStop™ Kernel operating system is running, the TSM software on the workstation communicates with the server through the TSM server software. See TSM client software.

TSM workstation. A PC-compatible workstation on which the Compaq Tandem Service Management (TSM) client software is running. The TSM workstations configured as the primary and backup dial-out points are referred to as the primary and backup system consoles.

unplanned outage. Time during which the system is not capable of doing useful work because of an unplanned interruption. Unplanned interruptions can include failures caused by faulty hardware, operator error, or disaster. Contrast with planned outage. See outage.

unstructured file. An array of bytes of data; it often contains text or program code.

VID. See volume ID number (VID).

ViewSys. An interactive utility that monitors system resources while the system is running.

volume. A physical disk pack mounted on a disk drive. A mirrored volume is a pair of physically independent disk packs that are configured and accessed as a single volume. One is the primary volume and the other is the mirror volume. All data written to one disk is written to the other as well. As long as both devices are operable, all data read from one disk could just as well be read from the other, because the content of both disks is the same.

volume ID number (VID). A string of up to six alphanumeric characters; for example, “AMBER1”, that is assigned to a tape during the MEDIACOM ADD TAPELABEL operation. This ID exists only for standard labeled tapes, and it resides on the tape’s first volume label and should also be on a hand-written label, which identifies the tape.

volume mode. The mode of operation in which Backup or Restore copies an entire disk volume of files.

VPROC. A utility program that displays product version information for object (file code 100) files.

WAN. See wide-area network (WAN).

WAN concentrator. See ServerNet Wide Area Network (SWAN) concentrator.

wide-area network (WAN). A network that operates over a larger geographical area than a LAN (typically, an area with a radius greater than one kilometer). The elements of a WAN may be separated by distances great enough to require telephone communications. Contrast with local-area network (LAN) and system area network (SAN).

$ZSVR. A labeled-tape server process. It is an optional product, without which you can perform only unlabeled-tape operations. When an application tries to open a labeled tape, it sends an open request to $ZSVR. $ZSVR checks to see whether the requested tape is mounted, and if it is not, $ZSVR sends an operator message to alert operations staff to mount that tape.

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

Z

Special Characters