UNISYS A Series

System Operations Guide

Priced Item

July 1992
Printed in U S America
8600 0387-100

UNISYS

Product Information Announcement

○ New Release ● Revision ○ Update ○ New Mail Code

Title

A Series System Operations Guide

This Product Information Announcement announces a revision of the *A Series System Operations Guide,* dated July 1992, relative to the Mark 4.0.0 System Software Release.

This guide explains the basic concepts and tasks involved in operating a Micro A, A1, A2, A3, A4, A5, or A6 system on a daily basis. This guide includes a part on basic concepts and system commands,and a part on operator tasks. This guide is written for beginning A Series system operators.

This revision contains updates to commands and terminal displays that affect A Series system operations.

Retain this Product Information Announcement as a record of changes made to the basic publication.

To order additional copies of this document

- United States customers call Unisys Direct at 1-800-448-1424
- All other customers contact your Unisys Subsidiary Librarian
- Unisys personnel use the Electronic Literature Ordering (ELO) system

Announcement only:

Announcement and attachments:
AS140

System: A Series
Release: Mark 4.0.0 July 1992

Part number: 8600 0387–100

UNISYS A Series

System Operations Guide

Release Mark 4.0.0

Priced Item

July 1992
Printed in U S America
8600 0387–100

Correspondence regarding this publication may be forwarded using the Product Information card at the back of the manual, or may be addressed directly to Unisys, Technical Publications, 25725 Jeronimo Road, Mission Viejo, CA 92691-2792.

Page Status

Page	Issue
iii	–100
iv	Blank
v through xv	–100
xvi	Blank
xvii	–100
xviii	Blank
1–1 through 1–13	–100
1–14	Blank
2–1 through 2–8	–100
3–1 through 3–15	–100
3–16	Blank
4–1 through 4–13	–100
4–14	Blank
5–1 through 5–9	–100
5–10	Blank
6–1 through 6–8	–100
7–1 through 7–4	–100
8–1 through 8–220	–100
Glossary–1 through 16	–100
Bibliography–1 through 2	–100
Index–1 through 22	–100

Unisys uses an 11-digit document numbering system. The suffix of the document number (1234 5678–*xyz*) indicates the document level. The first digit of the suffix (*x*) designates a revision level; the second digit (*y*) designates an update level. For example, the first release of a document has a suffix of –000. A suffix of –130 designates the third update to revision 1. The third digit (*z*) is used to indicate an errata for a particular level and is not reflected in the page status summary.

About This Guide

Purpose

Keeping your A Series system running smoothly depends on knowing how to monitor system activity and how to intervene when necessary. This guide will help you understand the basic concepts of operations and will show you how to perform fundamental tasks so you can quickly learn how to operate your A Series system.

Scope

This guide is divided into two parts so that you can easily find information when you need it. Part 1 presents concepts that are essential to operating your system efficiently and understanding the material presented in the remainder of the guide. Part 2 explains how to perform tasks that are fundamental to the operation of all Unisys systems. Commands are presented with only their basic elements so that you can learn them quickly. If you need a detailed explanation of a command as well as a complete list of its options, you can look up the command in the *A Series System Commands Operations Reference Manual*.

Audience

If you will be responsible for operating an A Series system on a daily basis, you should read this guide. If you are inexperienced with either mainframe computers or Unisys A Series systems, you will find this guide to be particularly beneficial.

Prerequisites

Before reading this guide, you should read the *A Series Systems Functional Overview* to understand the hardware and software components of A Series systems. You should also read the *A Series Documentation Library Overview* to learn where to look for more detailed information on products or topics.

How to Use This Guide

You should read the sections in Part 1 of this book sequentially to gain a basic understanding of A Series concepts and operations. Part 2 is intended as reference material and is designed for random access of information. Use Part 2 when you need step-by-step instructions for performing a variety of basic tasks.

When other books are referred to for the first time, the full title is given in italics. In subsequent references, the short version of the book title is used.

Organization

This document is divided into two parts. In addition a glossary, bibliography, and an index appear at the end of this document.

Part 1. Basic Concepts

This part presents the concepts necessary for a basic understanding of the operation of an A Series computer system.

Section 1. Understanding Basic Concepts

This section describes the basic hardware and software components that make up an A Series system, as well as where you can go for more advanced information regarding A Series operations.

Section 2. Performing Basic System Operations

This section provides information on entering commands, on the types of commands you can use, on the important master control program (MCP) options to set, and on displaying the status of system peripherals.

Section 3. Monitoring and Controlling Processes

This section describes the different types of processes found on A Series systems, how to monitor these processes, and how to control them.

Section 4. Responding to Waiting Processes

This section discusses how to identify processes that require operator intervention, how to determine what to do, and how to get waiting processes running again.

Section 5. Writing WFL Jobs That Execute Programs

This section describes how to create, test, and use simple work flow language (WFL) jobs to run daily and weekly programs, and how to create and use job queues to control job flow to the system processor.

Section 6. Managing Disk Space, the System Log, and Printing

This section describes how to check available disk space and copy disk directories, how to use the system log and control where the log is written, and how to deal with printer backup files.

Part 2. Operator Tasks

This part contains reference information describing how to do basic operator tasks.

Section 7. Using Task Procedures

This section introduces Part 2, describes how the part is organized, and explains how to use the task procedures.

Section 8. Task Procedures

This section contains a listing of basic procedures an operator performs periodically. The procedures consist of step-by-step instructions for performing the tasks.

Related Product Information

***A Series Documentation Library Overview* (form 8600 0361)**

This overview describes the library of A Series software documentation. It also provides an explanation of titling conventions, the procedure for ordering documentation, and an introduction to online documentation and its role in A Series product documentation. This overview is written for all users of A Series systems.

***A Series Print System (PrintS/ReprintS) Administration, Operations, and Programming Guide* (form 8600 1039)**

This guide describes the features of the Print System (PrintS) and provides a complete description of its command syntax. This guide is written for programmers, operators, system administrators, and other interactive users of Menu-Assisted Resource Control (MARC) and CANDE.

***A Series System Administration Guide* (form 8600 0437)**

This guide provides the reader with information required to make decisions about system configuration, peripheral configuration, file management, resource use, and other matters related to system administration. This guide is written for users with some, little, or no A Series experience who are responsible for making decisions about system administration.

***A Series System Commands Operations Reference Manual* (form 8600 0395)**

This manual gives a complete description of the system commands used to control system resources and work flow. This manual is written for systems operators and administrators.

A Series Systems Functional Overview **(form 8600 0353)**

This manual presents an overview of the A Series systems and serves as a central source of information for these systems. This overview is written for both new and experienced users of A Series systems, and for anyone wanting an introduction to these systems.

A Series Work Flow Language (WFL) Programming Reference Manual **(form 8600 1047)**

This manual presents the complete syntax and semantics of WFL. WFL is used to construct jobs that compile or run programs written in other languages and that perform library maintenance such as copying files. This manual is written for users of A Series systems who have some programming experience and who know how to create and edit files using CANDE or the Editor.

Contents

Section 2. Performing Basic System Operations

Section 3. Monitoring and Controlling Processes

Section 4. Responding to Waiting Processes

Figures

Tables

Part 1
Basic Concepts

Section 1
Understanding Basic Concepts

Your A Series computer system is made up of a variety of hardware and software components. To effectively operate your system, you must have a general understanding of what these components are and how they work together. This section introduces the essential groups of hardware and software and explains the basic concepts of files and file storage. The last part of this section shows you where to look for information on subjects not covered in this guide and suggests what documentation you should have in your operations library.

A Series Hardware Overview

Hardware is physical computer equipment, such as circuits, circuit boards, power supplies, cables, line printers, and terminals. All A Series systems require a minimum amount of hardware in order to function. This hardware can be divided into groups, called *subsystems,* according to the function each group performs. An A Series computer has four hardware subsystems:

- Processor
- Memory
- Input/output (I/O)
- Maintenance

The following topics explain the purpose and responsibilities of each subsystem.

Processor Subsystem

The processor subsystem consists of one or more processors that operate together to execute application and system programs. The main processing component is called the *central processor*. The central processor performs logical processing for all programs executed on the system. Some systems can be configured with multiple central processors.

Memory Subsystem

The memory subsystem consists of one or more memory storage boards and memory control *modules* (multiple boards connected together). This subsystem provides temporary storage of data and programs while they are in use and handles all transfers of data between main memory and the central processor. In general, the greater the memory capacity of a system, the greater the amount of information that can be processed at a time (up to processor and I/O limits).

Input/Output Subsystem

The I/O subsystem controls the communication between the central processor and all peripheral devices. (Peripheral devices include hardware such as disk units, tape drives, and printers.) Each device or group of devices is connected to a data link processor (DLP). A DLP consists of one or more circuit boards and serves as an interface between the system and a peripheral device, peripheral controller, or data communications line. Usually there is a specific DLP for each type of peripheral device. You can get information about the configuration of the DLPs in your system by entering the GC (Group Configuration) and SC (System Configuration) system commands discussed in Section 8 under "System."

Disk Devices

One of the most commonly used peripheral storage devices in the I/O subsystem is the magnetic disk. A magnetic disk is a flexible or hard platter whose surface is divided into concentric circles similar to the way an archery target is divided. These circles are called tracks, and each track is logically divided into parts called *sectors*. Data and code are stored in sectors.

The primary type of disk device used on A Series Systems is the hard disk. You use the hard disks built in to your system cabinet and any separate disk units you might have for storing and retrieving data. Some systems require the use of flexible disks (also referred to as *floppy disks, floppies,* or *diskettes)*for loading processor microcode and performing maintenance tasks.

Tape Drives

Another commonly used peripheral device is the magnetic tape drive. You use magnetic tapes mostly for long-term or *archival* storage, for backing up files, and for loading new programs onto the system. Some tape drives use open reel tapes, while others use quarter-inch or half-inch cartridge tapes. Two of the most common open reel tape recording formats are group coded recording (GCR), which has a data storage capacity of 6250 bits per inch (bpi), and phase encoded (PE), which has a data storage capacity of 1600 bpi. Not all tape drives are compatible with all formats.

Data Comm Subsystem

The I/O subsystem also includes the group of devices that are usually physically removed from the computers in the operations center and are connected to the main computer through data communications lines. This group of devices constitutes the *data comm subsystem* and includes remote terminals, printers, and data comm processors.

Data comm processors are specifically designed to control data communications and take some of the workload from the central processor. An I/O processor interfaces the data comm processors with the central processor and memory. The most common data comm processors are

- Data communications data link processor (DCDLP)
- Network support processor (NSP) - line support processor (LSP) combination
- CP 2000 communications processor
- Communication processor data link processor (CPDLP)
- Extended data communication data link processor (EDCDLP)

Maintenance Subsystem

The maintenance subsystem serves several purposes. It is the interface through which you can *initialize* (start), configure, or halt the system. This subsystem also can be used by a customer service engineer for diagnosing hardware and software problems from a remote location. For more information on using the maintenance subsystem, see the installation guide for your system type.

A Series Software Overview

While hardware consists of tangible objects, software consists of programs. Software directs the operation of the hardware and guides the computer system through the completion of a job or the solution of a user's problem. The software is stored as data in logical structures known as files. Files, in turn, are stored on tapes, disks, and in memory (during processing). Three general types of software run on A Series systems: system software, environmental software, and application software.

System Software

The system software coordinates system operations and manages the use of system resources. The system software consists of three specific types of software: the operating system, utility programs, and compilers. These are described in the following text.

Operating System

The operating system consists of a large program called the Master Control Program. We use the more familiar term *operating system* to represent the term *MCP* in this book. The operating system is the most important program on your system because it controls, organizes, and executes all system functions. The operating system has priority and authority over all other programs. Some of the most critical responsibilities of the operating system include

- Accepting commands you enter at the operator display terminal (ODT) and returning responses
- Initiating programs and monitoring their performance
- Managing input and output functions
- Assigning peripheral devices to programs

- Locating files for program input or output
- Maintaining system status and history
- Recognizing changes in peripheral status
- Recognizing and recording new files and new programs running on the system
- Monitoring the amount of memory and its location
- Allocating and deallocating memory

An important aspect of the operating system is that it has 47 options (commonly referred to as *MCP options*) that you can change. You can set, reset, or adjust these options to customize the way the operating system functions. Your system administrator determines how to configure these options to best serve the needs of your site. Knowing which options are set at your site, and what these options do, can help you understand how the operating system works. You can find a list of the operating system options, and instructions on how to display the option status on your system, under "Master Control Program (MCP)" in Section 8. Complete descriptions of all the options are in the *A Series System Commands Operations Reference Manual.*

Utilities

Another part of the system software consists of utilities. Utilities are programs that are designed to perform frequently required tasks, such as analyzing the system log. You can run utilities to perform many of your daily tasks. Also, utilities can be called by other programs when they require certain functions to be performed, thus saving the programmer from writing the same instructions over and over again. Some of the utilities you might use are the following:

Utility	Description
SYSTEM/DUMPALL	Generates a printout of files, and enables files to be copied from one medium to another
SYSTEM/FILECOPY	Copies multiple files from a disk to a tape or from a disk to another disk and gives you many file selection options not provided by the WFL COPY statement, such as copying all files created since a particular date
SYSTEM/FILEDATA	Prints reports about disk files, including a list of file attributes, a list of object code files and their attributes, a map showing file storage layout, and a disk checkerboard showing the space around files

For information about using the preceding utilities, see the *A Series System Software Utilities Operations Reference Manual.*

Utilities can also refer to collections of common *procedures* (sequences of instructions) called *libraries*. A library is a program that usually contains procedures, such as data conversion routines or mathematical functions. Programs running on the system can use library procedures when they require certain functions to be performed, which simplifies program creation and maintenance. Some libraries come with your system and are referred to as *system libraries*. Programmers and users on your system can develop custom libraries for your site, which are usually referred to as *support libraries*.

Compilers

Compilers constitute the portion of the system software responsible for translating the instructions written in a programming language (*source code*) into machine language (*object code*). The machine language version can then be read and executed by the hardware. All programs must be translated into object code by a compiler before they can be executed on your system.

Environmental Software

Environmental software establishes the environment or conditions under which application programs run. Two of the most common environmental software packages are Command and Edit (CANDE) and Communications Management System (COMS). Both CANDE and COMS are message control systems (MCSs) that handle data transmission between terminals and the operating system. You probably have both of these systems running on your system. Section 3 shows you how to display information about your message control systems.

Application Software

Application software consists of programs written to perform specific functions, such as performing spreadsheet and other accounting functions for a bank. Your site might purchase application programs or might use applications developed by its own programmers.

Basic File Concepts

A file is a basic unit for storing and accessing information. It is a named group of related records usually located on disk or tape. The following topics discuss basic file concepts. For more detailed information, see the *A Series I/O Subsystem Programming Guide.*

Logical and Physical Files

This book uses the word *file* for two different but related concepts, logical files and physical files. A physical file is the file as it is stored on a recording medium, such as a disk or tape. A logical file is a structure declared within a program. When a program executes an OPEN operation for a logical file, the system connects that logical file to a physical file on a peripheral I/O device. The program can then read data from and write data to that physical file. The program uses file attributes to control the selection of the physical file, establish properties of the logical and physical files, and test the status of the logical and physical files. The file name, maximum record size, and creation date are examples of file attributes.

Both the logical and physical files have file names. The physical (external) file name is recorded in a disk file header for disk files and in a tape label for tape files. The logical (internal) file name is a file variable declared within a program and known only within that program or within programs to which it has been passed as a formal parameter. More than one logical file can interact with the same physical file, and one logical file can interact with multiple physical files.

File Attributes

File attributes are specifications included with each file that define basic information about that file. Some of these attributes identify the file, while others describe the structure of the file. File attributes that identify the file include FILENAME, CYCLE, and VERSION. File attributes that describe the structure of the file include AREAS, AREALENGTH, BLOCKSIZE, and MAXRECSIZE. Many of the file attributes for a disk file are stored in the file header.

The most important file attribute is KIND, which determines whether the file is a disk file, tape file, remote file, printer file, and so forth.

When you want to create a new file, you must decide the values to assign to the file attributes. You can assign file attributes either within a program or with file equations when the program is compiled or run.

Resident and Nonresident Files

Disk files are often referred to as resident or nonresident. A file is resident if it is the primary copy of the file (as opposed to a backup copy) and it is stored on disk, regardless of whether the disk is online. A file is nonresident if it is stored only on a backup tape or if the file is a backup copy that is stored on another disk family. The terms resident and nonresident in this guide do not pertain to the file attribute RESIDENT.

Temporary and Permanent Files

Disk files can be either temporary or permanent. The header of a temporary file is not stored in the system directory of the family on which the file is located. When a program closes a temporary file, the contents of the file are no longer available, and the disk space of the file is returned to the system. A file is permanent when it has a disk file header in the system directory of the family on which the file is stored. When a program closes a permanent file, the contents of the file remain stored on disk.

Remote Files

A remote file is a special type of physical file used by most programs to pass information between a program and a remote device, such as a terminal. The program writes data to and receives data from the remote file, treating it as it would a local peripheral device. The file is attached to the remote device by the message control system (MCS). Remote files are named according to the conventions for disk file names, discussed in "Disk File Naming Conventions" in this section.

Disk Organization

On A Series systems, physical disks can be grouped into logical units called *families*. Each family is treated as a single unit by the system, even though the family might have more than one physical disk in it. All the disks in a family share the family name, which the system administrator usually assigns. The family name has 1 to 17 alphanumeric characters, and typically identifies the purpose of the files stored on that family. For example, a banking institution might put two disks into a family named ACCOUNTS to hold all its accounting programs and data. The family name and relevant information about the disk are stored on the first 28 sectors of the disk in a structure called a *label*.

To help distinguish among the disks in the family, the system assigns a *family index* number to a disk when it is first added to a family. Family index numbers range from 1 through 255.

The following text describes base and continuation disks, the system directory, and the halt/load family.

Base and Continuation Disks

The disk in a family that contains the system directory currently being used by the system is called the *base disk*. All other disks in the family are called *continuation disks*, and are marked as such.

System Directory

Each family has a base disk, which contains a directory that the system uses to access files. This system directory contains a *file header* for each permanent file stored on the family. The file header contains the name of the file, pointers to the physical locations of the data on the disk, and the attributes of the file. The file name for the system directory is SYSTEMDIRECTORY/nnn, where nnn is the family index number of the base disk on which the directory is stored. Section 6 provides detailed information on the system directory.

To avoid the possibility of losing access to files, you should copy the system directory to other disks in the family. You can make two copies, for a total of three system directories for each family. The system uses the copy located on the base disk. See "Creating Duplicate Directories (DD)" under "Disks" in Section 8 for instructions on copying this file.

Halt/Load Family

The family that contains the operating system and related files is called the *halt/load family*. The halt/load family can be any family, although it is often DISK.

A halt/load is a system initialization procedure that can be used when the system is not functioning properly and all attempts to correct it are not successful. A halt/load temporarily halts the system, clears memory, and then reloads the operating system.

See "Halt/Load, Performing (??PHL)" in Section 8 for instructions and additional information on performing a halt/load.

Disk File Naming Conventions

The following subsections describe the disk and disk file naming conventions that you need to know for operating your system. For more information see the *A Series Disk Subsystem Administration and Operations Guide.*

File Names

Each file has a unique name that distinguishes it from every other file. A file name consists of parts, called *nodes*, separated by a slash. A file name can have from 1 to 12 nodes. The following file name has three nodes:

```
NODE1/NODE2/NODE3
```

Each node begins with a number or letter and consists of up to 17 numbers and letters. Hyphens (-) and underscores (_) can be used as characters in the node. Any other characters require special processing by the system, so they are avoided. If you have to create a file node that includes special characters, enclose the entire node in quotation marks. In the following example, the period in the second node is a special character:

```
NODE1/"NODE.2"
```

Directory and Subdirectory Names

When there is more than one node in a file name, the first node is called a *directory*. Additional nodes might also be called directories or *subdirectories*. The last node is the name that identifies the specific content of the file and makes the file name unique.

The following file names are for an imaginary operations summary program. The first node places all the files under the directory OPSSUM. The second node, or subdirectory, provides more specific information about the files; that is, some were created on Monday and some on Tuesday. The last node identifies the content of the file, for example, whether it has data on charges, users, or system usage.

```
OPSSUM/MONDAY/CHARGES
OPSSUM/MONDAY/USERS
OPSSUM/MONDAY/USAGE
OPSSUM/TUESDAY/CHARGES
OPSSUM/TUESDAY/USERS
OPSSUM/TUESDAY/USAGE
```

Usercoded and Nonusercoded File Names

In addition to having nodes, a file created by a user on the system has that person's *usercode* attached to the beginning of it. A usercode is a name that identifies who owns a file and thus, who has access to it. Files that do not belong to any particular usercode have an asterisk (*) attached to the beginning of their names. These files are nonusercoded files, and can be read or copied by anyone with a valid usercode on the system where the file is resident.

The following examples show both a file with a usercode and one without:

```
(SMITH)NODE1/NODE2/NODE3
*NODE1/NODE2/NODE3
```

Family Names

Another piece of information that you might see in a file name is the *family name*, which is the name of the disk (or disks grouped logically as a single unit) where the file is stored. You can easily recognize the family name, because it is always preceded by the word ON. In the next example, OPERATIONS is the family name:

```
(SMITH)OBJECT/X ON OPERATIONS
```

File Titles

A file name that includes the family name is known as a *file title*. Some commands require the file name, while others require the file title, so it is important to remember the difference.

Tape File Concepts

The following topics cover the basic tape file concepts that you need to operate your system.

Labeled and Unlabeled Tapes

A Series I/O subsystems accept either labeled or unlabeled magnetic tapes. Tape labels are special records at the start, end, and between the files of a reel of tape that provide details about the files stored on the tape. The I/O subsystem recognizes and reads all standard tape labels of Unisys A Series, B 1000, B 7900, and V Series systems. The subsystem also reads ANSI Level 2 standard labels and IBM® 360-format labels. In addition, the subsystem can read unlabeled tapes. If a tape is unlabeled, files can be assigned to it only with operator intervention.

Tape File Names

Most of the tapes you deal with as an operator are library maintenance tapes. Library maintenance tapes are created by library maintenance, which is the software that

IBM is a registered trademark of International Business Machines Corporation.

copies files from disk to disk, disk to tape, tape to disk, and tape to tape. (An example of a non-library maintenance tape is a data tape created by a program.) Library maintenance tape file names have two levels. The first level is the name of the tape volume; the second level identifies the sequential number of the file on the tape. These tape file names have the following form:

```
<file name>/FILE<nnn>
```

The last three digits are incremented by one for each file in sequence. The first file on a library maintenance tape is a tape directory. This file has the identifier FILE000. For example, if a tape named OPSTAPE has three files on it, the file names would look similar to this:

```
OPSTAPE/FILE000
OPSTAPE/FILE001
OPSTAPE/FILE002
OPSTAPE/FILE003
```

Tape Serial Numbers

In addition to a tape name, tapes also have a serial number that provides identification of each individual tape volume. The serial number can consist of up to six numbers or alphabetic characters, or a combination of both. You can find information on assigning tape serial numbers under "Tapes" in Section 8. Some example serial numbers follow:

```
MYTAPE
52089
TEST1
89FILE
1DATA4
```

Tape File Storage

The I/O subsystem can store files in any of the following ways:

- Multiple files on a single tape
- One file split across multiple tapes
- Multiple files stored on a number of tapes, with files split across tapes as necessary

Library maintenance tapes often take the third form.

Where to Go for More Information

This guide attempts to give you all the basic information you need to run your system. However, it would be impossible to cover every aspect of A Series operations in one book. Another book, the *Documentation Library Overview*, can help you determine where to look for information that reaches beyond the scope of this book. The next subsection also recommends some books for your operations library. The subsection also tells you where you can order any publications you do not have.

Another tool that can help you find information is a book called the *A Series Software Documents Master Index*. This book lists all the topics covered in A Series documentation and identifies the books that cover each topic.

Setting Up Your Operations Library

From time to time you will need to get information from other A Series documents. The following table lists the publications that are recommended for your operations library. Some of these publications arrived with your system. Other publications accompany software purchases or are special purchases themselves. Ask your Unisys representative about obtaining copies of these documents.

Table 1–1 contains titles and descriptions of books recommended for your operations library. The books are all part of the A Series library, so the term *A Series* has been left off the book titles in the table.

Table 1–1. Suggested Documents for Your Operations Library

Title	Description
Documentation Library Overview	Gives an overview of all A Series documentation and provides information on using online and printed documentation
CANDE Configuration Reference Manual	Gives instructions on installing, configuring, and controlling CANDE
CANDE Operations Reference Manual	Provides instructions for initiating tasks from CANDE and gives commands for interrogating the operating environment
Communications Management System (COMS) Operations Guide	Provides instructions for controlling sessions, windows, and the network and gives procedures for initialization, shut down, backup, and recovery through COMS
Disk Subsystem Administration and Operations Guide	Describes the concepts and operation of A Series disks and disk units
Interactive Datacomm Configurator (IDC) Operations Guide	Describes how to use this menu-driven utility to define and maintain the configuration of a data communications subsystem
Memory Subsystem Administration and Operations Guide	Describes the concepts and operation of the memory management module of the operating system
Menu-Assisted Resource Control (MARC) Operations Guide	Explains how to use the menu-driven MARC interface to operate an A Series system

continued

Table 1–1. Suggested Documents for Your Operations Library (cont.)

Title	Description
Operating System Installation Guide	Describes how to use the UTILOADER and LOADER programs to install the operating system when a system is first initialized or when the operating system is not functioning.
Print System (PrintS/ReprintS) Administration, Operations, and Programming Guide	Describes the Print System (PrintS) and the Remote Print System (ReprintS) and explains how to use them
Printing Utilities Operations Guide	Describes the utilities that control the storage and printing of backup files
Security Administration Guide	Gives the system administrator the necessary information to establish security policy and configure system security to provide a level of protection appropriate for the uses of the system
Security Features Operations and Programming Guide	Describes the A Series security features and gives instructions for using each feature
Software Documents Master Index	Lists all the topics covered in A Series documentation and identifies the documents that cover each topic
Software Release Installation Guide	Describes how to use the Simple Installation program to install a new software release on an established A Series system
System Commands Operations Reference Manual	Describes the system commands that you can enter at the operator display terminal (ODT) and gives the complete syntax, as well as variations and examples, of each command
System Software Support Reference Manual	Describes methods for monitoring system resource utilization using information contained in the SYSTEM/LOG
System Software Utilities Operations Reference Manual	Describes the utility programs for performing operations such as copying, comparing, and merging files
Systems Functional Overview	Gives an overview of A Series functionality including advantages and benefits of A Series architecture

continued

Table 1–1. Suggested Documents for Your Operations Library (cont.)

Title	Description
Task Attributes Programming Reference Manual	Describes all the task attributes available on A Series systems, and gives examples of statements for reading and assigning task attributes in various programming languages. systems
Work Flow Language (WFL) Programming Reference Manual	Describes the syntax and semantics of WFL, a language that lets you write programs to control the running of tasks

Section 2
Performing Basic System Operations

Using the System Control Terminal (SCT)

Your main tool for communicating with your system is the system control terminal (SCT). This terminal has a video display unit and a keyboard and is designed especially for communicating with your hardware and software. The SCT has three distinct terminal modes:

- Operator display terminal (ODT) mode
- Maintenance display terminal (MDT) mode
- Remote display terminal (RDT) mode

You primarily use the ODT, which is your interface to the system software. From the ODT you can enter commands to monitor and control the system operating environment as well as the devices and programs used on the system. Other terminals can be configured to function as ODTs; however, these terminals do not have MDT or RDT functionality.

The MDT accesses the hardware that comprises the maintenance subsystem. From the MDT you can use a series of menus to start, halt, or configure your system. You can also use the MDT to test components of the system if you are having hardware problems. For more information about using the MDT and its menus, refer to the software installation guide or the operating guide for your particular system.

The RDT also accesses the maintenance subsystem of your computer, but it is designed to be used by a Unisys customer engineer for diagnosing hardware and software problems from a remote location. For more information about using the RDT, refer to the software installation guide or the operating guide for your particular system.

The system control terminal also gives you access to Menu-Assisted Resource Control (MARC), a menu-driven interface to the operating system. Refer to "Entering Commands from MARC Menus" in this section for information on using MARC.

Entering a Command at the ODT

You enter all commands at the ODT from the *home* position. The home position is usually the upper left corner of the screen, although the terminal you are using for the ODT might be configured differently. To place the cursor in home position, press the Home key. The location of this key depends on the type of terminal you have. Ask your supervisor to tell you where the Home key is if you do not know. The ODT might be in *forms mode*, which means a menu (a form) is displayed on the screen. This form might temporarily change the home position.

You can make sure your SCT is not in forms mode, or you can cause your SCT to exit forms mode by pressing the following keys:

```
CTRL-Q
```

That is, press the control key (usually labeled CTRL), and then press the Q key. Again, the exact location of the control key depends upon the type of terminal you have. Ask your supervisor if you do not know. This key sequence puts your terminal in *command mode*, and places the cursor in the home position. Once your cursor is in the home position, you can enter a command as shown in the following procedure.

To enter a command to the system, perform these steps:

1. Move your screen cursor to the home position (the upper left corner of the terminal screen).
2. Type the command.

   Note: *If your site uses the ETX (end-of-text) key, press the ETX key before pressing the XMIT key. See "Turning Off ETX Character Mode" in this section for more information about ETX mode.*
3. Press the XMIT (transmit) key.

The terminal your site uses as the system control terminal (SCT) might have different key functionality or a different configuration from that of the terminal used in this example. Refer to the user documentation for your terminal if you are having trouble finding the keys mentioned in this book.

Example

Try entering the TD (Time and Date) command for practice. Note that *enter* in this book means to type in a command at the ODT and then press the XMIT key. The TD command shows the time and date set at your system. Follow these steps:

1. Bring cursor to the home position on the screen, the upper left corner.
2. Type *TD*.
3. Press the XMIT key.

The time and date appear at the top of your screen.

Displaying and Clearing a System Response

If the system response to your command fills more than one screen, the letters NS (Next Screen) appear in the home position. To see the rest of the response, place your cursor after NS and press XMIT. If you do not want to see the rest of the response, enter the CQ (Clear Queue) command. This command clears all the system responses and messages queued for display.

Turning Off ETX Character Mode

End-of-text (ETX) mode used to be required to enter commands at the ODT. Pressing the XMIT key in ETX mode transmits all characters on the screen from the cursor to the ETX symbol.

ETX mode is no longer required for command entry on certain types of terminals. If ETX mode is active on your system, the ETX symbol appears at the bottom left or bottom right corner of your screen. Ask your supervisor if your terminal requires the ETX character. If not, you can simplify command entry by turning off ETX mode. Follow these steps:

1. Bring cursor to the home position on the screen (upper left corner).
2. Type

   ```
   TERM DCSTATION=TRUE
   ```
3. Press the ETX key.
4. Press the XMIT key.

This setting remains even when the system is stopped and restarted, but not if you turn off the terminal. After eliminating ETX mode, enter commands as instructed previously in “Entering a Command at the ODT.”

Using Commands

You can enter four main types of commands at the ODT: system commands, primitive commands, Work Flow Language (WFL) statements, and message control system (MCS) commands. These commands are described in the following paragraphs.

System Commands

System commands communicate with the operating system software, which controls the overall activity of your computer. With system commands you can take a disk offline and purge a tape, start and suspend a program, and perform many other functions. Most of the procedures that you perform day to day involve using system commands. Part 2 of this guide describes the system commands that are used most frequently in the basic procedures. Each command is in capital letters, followed by the command name in parentheses, like this: TD (Time and Date) system command. Subsequent references to the command use just the command: the TD command. You can find the entire set of system commands listed alphabetically in the *System Commands Reference Manual.*

Primitive Commands

When your system is very busy, it might take longer than usual to get a response after you enter a system command. There might also be a time when a software failure can make it impossible for you to enter system commands. If either of these situations occurs, you can enter certain commands as *primitive* commands. Primitive commands bypass the software that handles system commands and go directly to the operating system. To enter a primitive command, you must precede it with two question marks. For example, the primitive command that shows you all the processes that are waiting for a resource or for operator intervention is

```
??W
```

Primitive commands are listed with various procedures throughout this guide. You can find the complete set of primitive commands in the *System Commands Reference Manual.*

Work Flow Language (WFL) Statements

WFL is a Unisys language used to construct jobs that run one or more programs. WFL language constructs act like commands, but they are referred to as WFL statements. You most often use WFL statements to start WFL jobs, to run utility programs such as SYSTEM/FILECOPY, and to perform basic operations such as copying and removing disk files. Section 3, "Monitoring and Controlling Processes," explains how to start WFL jobs. Section 5, "Writing WFL Jobs That Execute Programs," explains how to write WFL jobs. Other procedures involving WFL statements are presented in Part 2. For complete details about WFL, see the *WFL Reference Manual*.

Message Control System (MCS) Commands

An MCS is a program that controls the flow of messages between terminals, application programs, and the operating system. You will probably have the message control systems Communications Management System (COMS) and Command and Edit (CANDE) running on your system. From the ODT, you communicate with terminals and applications controlled by message control systems by using the SM (Send Message) command with the appropriate syntax. For example, you might need to send the users of CANDE and COMS a message to log off for system maintenance. See Part 2 for instructions on sending messages to message control systems.

Using Common Command Syntax Elements

Many commands require you to enter information such as a unit number or an abbreviation for a device. When there is information for you to supply, you see a *variable* in the command. A variable consists of one or more words surrounded by angle brackets (<>). The words between the brackets indicates the type of information you must supply with the command. In the following example, <device> is the variable:

```
PER <device>-
```

When you enter a command, replace the variable with a value appropriate to your needs, and retain all of the elements that are outside the angle brackets. For example, to assign a value to the command shown above, replace <device> with the standard abbreviation for any device (see Table 2–1).

To enter the preceding command using the abbreviation for a disk, PK, enter the following:

```
PER PK-
```

Note that these variables can apply to any commands you enter in ODT mode, from MARC, or from a message control system (MCS).

Some of the variables you see frequently in this guide are the following:

Variable	Explanation
<device>	A standard abbreviation for a peripheral device (see Table 2–1)
<family name>	A name assigned by your site to one physical disk volume or a group of physical disk volumes
<family index>	A number assigned to each member of a disk family
<mix number>	The number assigned by the operating system to a job or task when the job or task is initiated.
<unit number>	The hardware number assigned to a device. **Note:** *The <unit number> is static, that is, wired into the hardware. The other numbers here are dynamic, that is, set by software.*
<unit number list>	A series of unit numbers separated by commas
<serial number>	An identifier 1 to 6 characters long assigned by your site to each tape and disk volume

Note that anytime you see the word *list* as part of a variable, you can enter more than one of that particular item. Be sure to separate the items with commas. For example, if you see *unit number list,* you can enter more than one unit number by using a format like the following:

```
12,47,56
```

Table 2–1 shows the abbreviations you can use for the device variable.

Table 2–1. Device Abbreviations Used in System Commands

Abbreviation	Device Represented	Description
CP	Card punch	A machine that punches holes (data) into cards.
CR	Card reader	A machine that reads holes punched into cards as data.
DC	Data comm device	A processor that serves as the interface between a peripheral device and the system, either a network support processor (NSP), a line support processor (LSP), or a data communications data link processor (DCDLP).
PK	Disk	A device consisting of one or more circular platters stacked on a central spindle used for storing data. (PK stands for pack, which is an older term for disk.)
IP	Image printer	A laser printer that prints a complete page of images at one time.
LP	Line printer	A printer that prints one line at a time.
MT	Magnetic tape	A tape with a magnetizable surface on which data can be stored and from which data can be retrieved. The tape can be either a reel-to-reel tape or a cartridge tape.
SC	System control terminal (and ODT)	The terminal that lets you enter commands that query the system and send instructions to the system.

Entering Commands from MARC Menus

The previous topics explain how to perform system functions by entering commands directly from the ODT. You can also perform system functions from menus by using the Menu-Assisted Resource Control (MARC) software that comes with your system. MARC provides a menu interface, so you do not have to remember command syntax. In addition, MARC has help text to guide you in using commands and an on-screen glossary that you can use to look up unfamiliar terms. Because MARC provides full-screen menus, it does not have the same system monitoring capabilities as your ODT. For this reason, you might want to use the ODT most of the time and use MARC when you are first learning commands or when performing a task that you do not perform regularly.

Accessing MARC

To get to the MARC environment from the ODT, do the following:

1. Enter the following primitive command from the home position:

```
??MARC
```

   If the MARC log-on screen appears, skip to step 3.

2. Press the space bar and then press the XMIT key to display the MARC log-on screen.

3. Enter the usercode and password that log you on to the system. (You can get the usercode and password from your system administrator.) A menu similar to the one shown in Figure 2–1 appears:

```
         MARC - MENU-ASSISTED RESOURCE CONTROL            Ø3:47 PM
Action:
         HOme PRev GO PArent COmnd                  (Press SPCFY for Help)
-------------------------------------------------------------------------
       Session                Jobs and Tasks           System Software
TEACH  What is MARC?    IE     InfoExec         DMPS   Dumps
NDT    News Date Time   TOOLS  Application Tools LOG   Logging
BYE    Log Off          UTIL   System Utilities MEM    Memory Management
CANDE  Cande Window     RUN    Run a Task       USERS  Usercode/Password
ON     Change Window    START  Start a WFL Job  PS     Printing System
SC     Session Control  JQ     Job Queues       SYS    System Control
                        JD     Job Display
       Data Comm        JC     Job Control             Hardware
SEND   Send Messages                            IO     I/O Devices
COMS   COMS Displays           Files            MM     Memory Modules
BNA    BNA Commands     SP     Special Programs PROC   Processors
DC     DataComm Control LIBS   System Libraries OTHER  Other Devices
                        FILE   File Management  CONFIG System Config

Choice:
-------------------------------------------------------------------------
```

Figure 2–1. MARC Home Menu

Note that this menu can vary, depending upon the privileges of the usercode you used to log on to MARC.

You can make selections from the MARC menu by entering a MARC menu option (shown in capital letters) on the Choice line. You can bypass MARC menus and enter a variety of commands directly from the Action line. For information on installing and operating MARC, see the *MARC Operations Guide*.

To return to the ODT environment from MARC, enter BYE on the Choice line of the MARC home menu. The MARC log-on screen reappears. Then enter the following primitive command from the MARC log-on screen:

```
??ODT
```

Section 3
Monitoring and Controlling Processes

When the instructions in a computer program are executed by the system, the program becomes a *process*. The operating system assigns each process a "mix number." A large part of your job involves monitoring the processes that run on your system and making sure they have the necessary resources to be completed successfully. This section discusses the types of processes you might encounter and explains how to display and control processes with system commands.

Understanding Jobs and Tasks

There are two main types of processes: jobs and tasks. A job usually initiates tasks, and the tasks run under the control of the job. Tasks depend upon the continued existence of their job. If you terminate a job, its tasks are immediately terminated by the system.

You can recognize a job by looking for the word JOB before the file title when you display the active jobs in the system mix. Another characteristic of a job is that after it terminates, it produces a job summary documenting how it and its tasks ran. (The creation of job summaries is controlled by the JOBSUMMARY task attribute.)

Special Types of Processes

In addition to jobs, there are three special types of independent processes that you should be aware of:

- Message control systems (MCSs)
- Independent runners
- Libraries

The first two types of processes appear in your mix display, and you should be careful not to terminate them. Libraries do not appear in the mix, but you can display them by entering the LIBS ALL (Library Task Entries) command. These special processes are described in detail in the following text.

Message Control Systems (MCSs)

A message control system (MCS) is software that enables user terminals connected to the system through data communications lines to communicate with the operating system. The MCS controls the flow of commands and messages to and from user terminals, thus relieving the operating system of some of its work. You use MCS commands to affect user terminals, for example to fix a terminal that is not communicating with the system, or to find out the status of a terminal. The only way you can get commands to an MCS from the ODT is through the SM (Send Message) command described in "Sending Commands to an MCS or Database (SM)" in Section 8.

COMS and CANDE are two message control systems that you probably have running on your system. You can get information about MCS commands and procedures by referring to the *CANDE Operations Reference Manual* and the *COMS Operations Guide*.

You can display information about the MCSs running on your system by entering the MCS (Message Control System) command. A response similar to the following appears:

```
--Mix-Status----Tasks--Stations--Num----- MCS SUMMARY ------------------
 1234 ACTIVE     14      356       1  SYSTEM/CANDE
  354 ACTIVE     32      127       2  SYSTEM/COMS
      INACTIVE             3       3  SYSTEM/RJE
```

These fields are described as follows:

Mix
: Mix number of the MCS.

Status
: One of four values: ACTIVE, INACTIVE, DSED, REQUIRED.

- ACTIVE means the MCS is currently running.
- INACTIVE means the MCS is not running because it either terminated normally or was never executed.
- DSED means the MCS was discontinued abnormally.
- REQUIRED means an unsuccessful attempt was made to execute the MCS. The attempt was unsuccessful because the MCS was previously discontinued or the MCS code file was not found. This failed attempt causes an "MCS REQUIRED" message to appear at the ODT.

Tasks
: Number of active tasks that were initiated from that MCS.

Stations
: Number of logical stations controlled or owned by the MCS. (Stations include physical terminals and printers, as well as logical devices. When COMS and CANDE both run on a system, COMS considers dialogs to be logical stations.)

Num
: Number of the MCS.

MCS SUMMARY
: File name of the MCS.

Independent Runners

Independent runners are procedures that are initiated and used by the operating system to perform certain actions. For example, the procedure that reconfigures disks is an independent runner that appears in the job mix when you enter an RC system command.

Libraries

Libraries are sets of common, related procedures, such as mathematical routines, that programs can use. Many standard libraries come with your system. These libraries are called *system libraries*. Examples of system libraries are the GENERALSUPPORT library and the PRINTSUPPORT library. Other libraries can be developed by onsite programmers. These libraries are usually referred to as *support libraries*. Both system libraries (such as PRINTSUPPORT and GENERALSUPPORT) and support libraries are controlled by the SL system command. See the description of the SL command under "Libraries, Support (SL)" in Section 8. Programs are written so that they can use the procedures in these libraries. That way, the code for the procedures does not have to be included in the programs, and they run more efficiently.

There are three types of libraries from the operator's point of view, described in the following:

Library Type	Description
Permanent	The library is a permanent part of the mix. The library is said to be "frozen" in the mix.
Temporary	The library enters the mix when a process requests it and leaves when processes are no longer using it.
Control	The library enters and leaves the mix based upon criteria provided by a programmer.

If a permanent library is not being used very often by programs, it is possible to free up memory by making that library temporary. The following topics explain how to display library status and how to make permanent libraries temporary.

Displaying Libraries

You can determine which libraries are being used on your system at any given time by entering the LIBS (Library Task Entries) command. The output from this command looks similar to the following:

```
---Mix--Frz---Shr-Usr--------- 5 FROZEN LIBRARIES ----------------------
* 1908  Temp  All   1 SL *SYSTEM/TADSSUPPORT ON ACCOUNTS
* 1626  Perm  Run   2 SL *SYSTEM/COMS/UTILITY/FORMATS ON USERDISK
* 1595  Perm  Priv  1    JOB *SYSTEM/SCREENDESIGN/INTERFACE ON ENG179
* 1593  Temp  All   2    JOB *SYSTEM/SCREENDESIGN/FORMATS ON ENG179
  1485  Ctrl  All   5    (MACS)TEST/FREEZE/CONTROL
```

The asterisk (*) in the first column indicates that the entry is being displayed for the first time. The other fields provide the following information:

Mix	Shows the mix number of the library
Frz (Freeze)	Shows whether the library is permanent (Perm), temporary (Temp), or control (Ctrl)
Shr (Shared Specs)	Shows who is entitled to use this library by displaying one of the following values: • All (Shared by all processes) • Run (Shared by run unit which consists of a program and the libraries it is accessing) • Priv (Used by one process or family of processes only)

continued

continued

Usr (User)	Indicates how many users are currently accessing that library

After the number of users, some entries show the letters SL. This indicates that the library is a support library, which is a library that can be accessed by a function name, instead of the title of the library's code file. The remainder of the line shows the file name of the library.

Enabling Libraries to Leave the Mix

As discussed previously, permanent libraries remain in memory at all times. Libraries that are used frequently by many programs should remain permanent. However, if you see that certain frozen libraries are not used very heavily, you can make them temporary by using the THAW (Thaw Frozen Library) command described in the following text. A temporary library stays in memory until all programs finish using it, and then the system removes it from memory. When any program needs the library, the system reads it into memory again.

While making the most efficient use of memory, temporary libraries can be a burden on the system if they are constantly being read into memory from disk. Certain libraries might also be required to maintain system status information between calls; thawing and restarting such a library can cause bad side effects. For these reasons, you should always check with your system administrator before thawing any libraries.

To change a library from frozen to temporary, first get the mix number of the library with the LIBS command. Then enter the THAW (Thaw Frozen Library) command in the following format:

```
<mix number list> THAW
```

The library remains in the mix until all programs finish using it.

Understanding Process Priority

The operating system uses the concept of *priority* to determine how much of the processor that the processes running on the system have access to at any given time. Priority consists of two elements: the priority category and the priority assigned to each process.

The main priority ranking is based on category. Invisible independent runners (ones you do not see in the mix), message control systems (MCSs), control programs, and WFL jobs are all in special high-priority categories. User processes, which are all other processes, are in a low-priority category.

Within each category, the priority assigned to an individual process determines the relative priority of the process in that category. Thus, an MCS with a priority of 50 has a higher priority than an MCS of priority 40 and also a higher priority than a user process of priority 90.

The priority value ranges from 0 through 99. The default system priority is 50. A process executed with a priority of 0 runs at the default priority set for the usercode the process is running under or the default for the system (50) if the process is not running under a usercode. Within each priority category, a value of 1 is the lowest priority, and 99 is the highest.

You can see what priority a process is running under by looking at the ADM screen. The active, waiting, and scheduled entries all have a priority heading. If you are not using ADM, you can enter the MX (Mix Entries) command to display processes and their priorities.

You can change the priority of a process with the PR (Priority) system command, which is described in the *System Commands Reference Manual.*

Monitoring Processes at the ODT

When your system was started for the first time, your ODT probably came up in automatic display mode (ADM). ADM displays the status of processes on your ODT screen so that you can monitor system activity. The system updates the screen every 9 seconds, by default, showing the change in status of any process. If you have more processes than can be displayed in the given space, the system scrolls through the processes and displays a new set each time it updates the screen. The system marks all new entries with an asterisk (*) to help you identify those you have not seen before.

If ADM is active, your screen looks similar to the one shown in Figure 3–1:

```
------Mix-Pri-CPU Time----------- 5 ACTIVE ENTRIES ----------------------
   E 7226 80     14:17   (NORM) *DCALGOL (NORM)OBJECT/SYMBOL/NED
     7259 80     06:03   JOB *SYSTEM/CANDE ON PACK
     7262 80     01:07   *CANDE/STACK01
     7248 80     15:08   *SYSTEM/COMS ON PACK
     7253 50     14:12   MARC
------Job-Task-Pri--Elapsed-------- 1 WAITING ENTRY ---------------------
   * 4237/5663 50   11:14 (HENRY) (HENRY)OBJECT/SYMBOL/GOODUTIL ON TOP
               NO FILE OBJECT/SCREENDESIGN/USA ON TOP
------Mix-Pri---Elapsed------------ 1 SCHEDULED ENTRY -------------------
     5655  50     07:22 (SMITH) *DCALGOL (SMITH)OBJECT/PATCH/X
------Job-Task-Time--Hist----------- COMPLETED ENTRIES ------------------
   * 5653\5653 08:43 EOT   JOB (DON) *OBJECT/SPO/II
   * 5653\5654 08:42 EOT   (DON) *OBJECT/NOSUM
     5637\5652 08:40 EOT   (GEORGE) (GEORGE)OBJECT/SD ON FIRE
     5637\5651 08:40 O-DS  (MACS) (GEORGE)OBJECT/TAB ON FIRE
------Mix--Time-------------------- MESSAGES ----------------------------
   * 5633 08:42 PK51 (BARB)CANDE/TEXT640 CHANGED TO (BARB)PORTFILE ON FIRE
     5633 08:41 PK51 (BARB)PORTFILE REMOVED ON FIRE
```

Figure 3–1. Automatic Display Mode (ADM) Screen

If ADM is not active on your screen, you should start it by entering the following command from the ODT:

```
ADM
```

Note that if ADM is already active, the current settings are displayed when you enter ADM.

ADM shows the status of the *mix*, which is the collection of processes in various stages of processing on the system. ADM defaults display five categories of information:

Active Entries	Processes that are currently running.
Waiting Entries	Processes that are waiting for some resource, like a specific file or tape drive, or for operator input.

continued

continued

Scheduled Entries	Processes that have not begun processing for one of the following reasons: • There is not enough memory available at the present time. • The system needs more disk space for the system log or overlay files. • A CM (Change MCP) command has been issued and the operating system is going to be changed. • An HS (Hold Schedule) command was issued, which stops the initiation of new processes.
Completed Entries	Processes that have finished executing.
Messages	Any messages generated by processes throughout the various stages of execution on the system. A message might indicate that a process finished executing normally or did not finish for some reason.

Table 3–1 lists the items that appear on the ADM example screen:

Table 3–1. Process Status Information on the ADM Screen

ADM Screen Information	Meaning
* (in first column)	New entry.
E (in first column)	An error symbol that appears only when compilation processes encounter a syntax error. (A compilation process is a single run of a compiler, such as the ALGOL compiler.)
Mix number	An identification number assigned to the process when it enters the mix.
Job number	The mix number assigned to a job when it enters the mix. An asterisk (*) in this field indicates that the process is a task that is not associated with a job.
Task number	The mix number assigned to a task when it enters the mix. For a job, the task number is the same as the job number.
Pri	The priority of the process. (This number represents how quickly the process is run in relation to the other processes in the mix. The scale is 1 through 99; 99 is "fast." Priority is discussed earlier in this section.)
CPU Time	The amount of processor time an active process has used.
Elapsed	The amount of time that has passed since the process was initiated.
Time	The time a process took from start to finish.

continued

Table 3–1. Process Status Information on the ADM Screen (cont.)

ADM Screen Information	Meaning
Hist	(For completed entries only) Abbreviations for messages that indicate the history of a process execution. All messages ending in the letters DS indicate that the process was terminated abnormally. O-DS indicates that the process was discontinued by the operator. R-DS indicates that the process exceeded a resource limit. EOJ (end of job) and EOT (end of task) indicate that a process ended normally. SNTX appears only for compilations that ended normally but had errors in the program being compiled. All the other messages indicate that the program did something illegal or contradictory.
File title	The last block of information on the line, which shows the file title of the program code file that the process is executing, or the name of the process (in the case of compilers, the title of the compiler code file without the node SYSTEM, followed by the title of the output code file). file name the process is stored in. This block might also include the following: • The word JOB if the process is a job • The usercode in parentheses if the process belongs to a user Section 2 describes all the elements of a file name.
Messages	Information in message form that appears in two places in the ADM screen. A message appears after each waiting entry, indicating why the system stopped executing the process. Usually, to get a process out of the waiting entries and back into an active state, you have to enter a command. Section 4 describes the commands you can use to reactivate waiting entries. Messages generated by all processes throughout the various stages of execution appear in the messages category of the ADM screen.

Suspending and Resuming ADM

After you enter a command, the system response overwrites some or all of the ADM screen. The system response remains for the ADM delay time (9 seconds by default), and then the ADM screen overwrites the response and continues as usual. If you need to see the system response a few seconds longer, press the Local key (usually labeled LCL) before the screen is overwritten. When you want the ADM screen to return, press the RECEIVE key.

Sometimes it is convenient to stop ADM for a longer time, for example, if you know that a command response will be quite lengthy or if you have several commands to enter at the ODT. You can temporarily stop ADM by entering

```
ADM ST
```

You can resume the ADM display by entering

```
ADM OK
```

Customizing Your ADM Screen Display

By default, the ADM command allocates a certain number of lines on your ODT screen to each group of mix entries. Table 3–2 lists the default settings for each category of information. For most sites, these default settings are satisfactory. However, you can change the number of lines for a group of entries or include other ADM options to display additional system information. The following topics explain how to customize ADM.

Table 3–2. Default ADM Settings

Category	System Command Used to Display Process Status	Number of Lines Allocated (Includes Heading)
Active processes	A	7
Waiting processes	W	3
Scheduled processes	S	2
Completed processes	C	5
Messages	MSG	Remainder of screen (7 lines on a 24-line screen)

Changing the Number of Lines for Process Entries

You must allocate at least two lines for each category of processes you want to display on your ADM screen (the heading takes up one line). The total number of lines for all categories cannot exceed 25. To change the number of lines allocated to each group of entries, enter a command similar to the following, inserting the number of lines you want for each category:

```
ADM (A4, W5, S2, C6, MSG) DELAY 9
```

This command divides the ADM screen into the following portions:

Category	Number of lines
Active entries	4
Waiting entries	5
Scheduled entries	2
Completed entries	6
Messages	Remainder of screen

The *DELAY 9* specifies a delay time of 9 seconds between each display update. Note that a delay time of less than 9 seconds taxes system resources and slows the system down. You should not use delay times of less than 9 seconds.

To check your changes, enter *ADM*. The current settings are displayed at the top of your screen.

Displaying Additional Information

In addition to changing the number of lines allocated for each group of entries, you can also specify options that display additional system information. The following options can be very useful:

Option	Explanation
PER MT	Use PER MT if you mount tapes frequently. This option shows you the tape name, serial number, and other useful information.
PER LP	Use PER LP if you print many jobs on line printers. This option helps you keep track of the jobs printing on various printers.
JOBS	Use JOBS if you have fewer than 30 processes running on your system. This option shows you the relationship between processes, which is useful in determining what action to take if there is a problem.

It is not advisable to use PER PK as an ADM option because it requires a lot of system resources and can slow your system.

The information that appears for PER MT and PER LP is the same information that you get when you enter the *PER <device>* command. You can find a discussion of the output for this command in Section 2.

To use one of the preceding options in your ADM setup, enter a command similar to the following, including one or more options, with the number of lines you desire:

```
ADM (A7, W3, S2, C5, MSG 5, PER MT 2) DELAY 9
```

This command allocates lines to the five default ADM options, and assigns two lines to the PER MT option.

To check your ADM settings, enter *ADM*. The current settings are displayed at the top of the screen.

If you want to include all three options listed above with the default ADM settings, you must divide the ADM display over two screen pages. To do this, enter the ADM command in the following format:

```
ADM (A7, W3, C5, MSG) (S2, PER MT 3, PER LP 3, JOBS) DELAY 9
```

The first group of options in parentheses appears on the screen and remains for 9 seconds, then the next group of options in parentheses appears and remains for 9 seconds, then the first group reappears, and so on.

There are many other options you can set in your ADM display. You can find these options in the *System Commands Reference Manual.*

Returning to the Default Settings

To return to the default settings, do the following:

1. Stop ADM and discard the current ADM settings by entering:

   ```
   ADM-
   ```

2. Restart ADM with the default settings by entering:

   ```
   ADM
   ```

Note that the ADM command has two different effects according to when you use it. As shown previously, entering ADM after entering ADM– restarts ADM with the default settings. Entering ADM at any other time simply displays the current ADM settings.

Displaying Processes with System Commands

You might want to see all of the mix entries in a particular status group without being limited by the number of lines allocated for that group on the ADM screen. This is especially true for messages, as you sometimes need to see several messages to determine how a process is progressing.

You can enter one of the system commands listed in Table 3–3 to see all of the processes in a particular group. Note, however, that a process might be *suppressed.* Suppressed processes are not included in the displays unless you specify the ALL option as described in Table 3–3. Processes that are run often and typically do not need intervention can be suppressed so that information about them does not take up space in the system displays. Refer to the *System Commands Reference Manual* for information about the SUPPRESS (Suppress Display) system command.

Note that when you enter one of the commands for displaying process status, the system response remains on the screen for 9 seconds (the default ADM delay time). After 9 seconds, the screen returns to ADM. If you want to keep the system response on the screen longer, you can use the Local key, or you can temporarily stop ADM. See “Suspending and Resuming ADM” in this section.

To display only the processes for a particular usercode, enter a command from Table 3–2 with the desired usercode in the following format:

```
<command> USER <usercode>
```

For example, to see all active processes running under the usercode SMITH, enter

```
A USER SMITH
```

When you use the MSG command, the system only displays the 21 most recent messages. These messages might require more than one screen to display. You probably do not need to see all these messages unless you suspect a problem. To print a chronological listing of all the system messages that were generated during a certain time period, use the LOG (Analyze Log) command with the following syntax. You can get complete information on using the LOG command to read system messages and other information in the SYSTEM/SUMLOG file by referring to the *System Software Support Reference Manual*. When entering the following syntax, be sure to put all times in 24-hour format:

```
LOG <beginning time> TO <end time> MSG
```

For example, to get a listing of all the messages in the system log from between 8:00 a.m. and 7:00 p.m., enter

```
LOG Ø8:ØØ TO 19:ØØ MSG
```

The following table summarizes the commands for displaying process status on your system.

Table 3–3. Commands for Displaying Process Status

Command	Action
A	Displays active processes, except those suppressed with the SUPPRESS command.
A ALL	Displays all active processes, including those that have been suppressed with the SUPPRESS command.
C	Displays recently completed processes.
J	Displays active, waiting, and scheduled processes and shows the relationship between them.
J ALL	Displays suppressed processes in addition to the information displayed by the J command.
MSG	Displays system messages. You can display only one screen of messages at any given time.
MX	Displays active, waiting, and scheduled processes and shows the relationship between them, as well as any system messages.
MX ALL	Displays suppressed processes in addition to the information displayed by the MX command.
S ALL	Displays all scheduled processes.
<mix number> TI	Displays time information for a process, including the amount of processor time used and the amount of time that has elapsed since the process entered the mix.
W	Displays all waiting processes.
<mix number> Y	Displays the status of the specified process.

Suppressing Operator Message Displays at the ODT

Operator messages are displayed at the ODT by the operating system or by running tasks. Since the number of operator messages displayed at the ODT can be large, important messages can be lost in the quantity of routine messages constantly appearing on the ODT screen.

You can use the message suppression feature to suppress the appearance of selected system and job messages on the ODT and remote terminals. The message suppression feature can be initiated in several ways:

- You can enter MSC (Message Control) suppression and display commands directly from the Action line of any MARC menu or by using MARC menus.
- You can create a data file of MSC suppression and display commands that can be loaded as needed.

The following are the methods that you can use to remove the commands:

- Deactivate the commands individually with opposing commands.
- Enter a *CLEAR* command that removes the current MSC commands.

When a halt/load occurs, all the MSC commands are canceled. If you want to have a basic data file of MSC commands loaded after a halt/load, you can create a MSC default file. For directions for creating this automatic startup file, refer to the the *MARC Operations Guide.*

To learn more about the various tasks the MSC command can perform, as well as some simple examples of using MSC commands in MARC, refer to "Messages, Suppressing (MSC)," in Section 8. For a complete discussion of this powerful and complex feature, see the *MARC Operations Guide.*

Understanding File Attributes

Commands often require more information than previous examples indicate. Often the file attributes declared in a program, such as the physical file name and device type, do not match the actual file needed. You can use family substitution statements and file equations to modify the file attributes specified in a program.

A *file attribute* is an element that describes a characteristic of a file and provides information the system needs to handle the file. Examples of file attributes are the file title, record size, number of areas, and the date the file was created. Family substitution and file equation statements enable you to change file attributes when you run a program or Work Flow Language (WFL) job. The following topics explain how to use these special types of statements.

If a program or job needs substitutions, and you execute it without the correct statements, it is put in the waiting entries. Section 4, "Responding to Waiting Processes," explains how to supply the attributes to reactivate waiting entries.

Declaring the Correct Disk Family

If no family name is specified in the file title or family name of a program or job, the system looks for that file or creates the file on the DISK family. Typically only system software is stored on DISK, and data files and a variety of other files are stored on many different families. To tell a program or job where to find the files it needs, you can include a *family substitution* statement with the execution command. Usually the person submitting the program or job knows what family or families must be accessed.

The family substitution statement has the following format:

```
FAMILY DISK = ENGR OTHERWISE DISK
       ----   ----          ----
        |      |             |
     target    |             |
     family    |             |
           substitute        |
             family      alternate
                           family
```

If the family name for a logical file, or a WFL RUN, PROCESS, COPY, ADD, ARCHIVE, CHAnge or REMOVE statement matches the target family name, the system searches for that file on the substitute family. If it cannot find the file on the substitute family, it looks on the alternate family. If it does not find the file on the alternate family, the process might stop and appear as a waiting entry. (Not all file requests wait. You can reactivate processes that appear in the waiting entries list in the mix displays by entering a system command. Section 4 discusses the possible system command responses.)

For information on using family substitution when running a program, see "Programs, Running (RUN)" in Section 8.

Specifying Alternate File Attributes

Many A Series programs and compilers are written to use a generic file name and device kind with the assumption that you supply the physical file name and indicate the device on which the file is stored or should be created at run time. You can use an instructional sequence called a *file equation* to cause a program or job to look for an input file, or create an output file, or both, under a file name, on a family, or on a kind of device other than the one specified in the program.

To create a file equation, you must first know the *internal name* of the file as it was declared in the program. The programmers at your site should be able to give you the internal file names for any program they developed. For programs supplied by Unisys, the documentation often discusses the files used by the program and their internal names. Some programs are written to use a card reader file named CARD. Most sites need to change the file name as well as the device kind for these programs.

The attributes that you most commonly need to put in a file equation are as follows:

Internal file name	The logical name of the file given in the program or job (get the name from your programmer)

continued

continued

Kind	The device type where the file is actually stored, such as DISK or TAPE
File name	The external name of the file used in commands and displays
Family	The name of the disk family where the file is stored
Title	The external file name and the family name connected by ON, for example, PROG/TIME ON OPSPACK

Note that in addition to file equations, WFL provides a closely related feature called *task equations*. Whereas file equations affect file attributes, task equations affect task attributes, which affect the behavior of a particular job or task. Some common task attributes are usercode, priority, and family. See the *WFL Reference Manual* for complete instructions on using task equations.

For information about using common file equations, see "Programs, Running (RUN)" in Section 8.

Section 4
Responding to Waiting Processes

When a process needs some resource that is not available to it, the system suspends the process temporarily and puts it in the waiting entries. You can see which processes are waiting by monitoring the ADM screen (see Section 3, "Monitoring and Controlling Processes") or by entering the W (Waiting Mix Entries) command if you are not using ADM. If the process needs a resource, like a specific file or tape drive, it might resume automatically when the resource becomes available. Other processes are waiting on operator confirmation or input and do not restart without intervention.

Waiting processes that require operator intervention to continue executing display one of two types of messages: either an RSVP message, or an AX message. This subsection shows how to read and respond to RSVP messages.

Interpreting Messages Displayed by Waiting Processes

Each waiting process that requires operator intervention displays an RSVP message that indicates why the process stopped executing. RSVP messages are displayed beneath each entry as shown in the following Waiting Entries display:

```
--Job-Task-Pri---Elapsed----------- 2 WAITING ENTRIES -----------------
 7916/792Ø  5Ø     1Ø:23 (SUE)COPY/JOB ON ORDS
             SUE/FILEØØØ REQUIRES MT #1
 163Ø/1638  5Ø   1:1Ø:39 (BOBG)(BOBG)OBJECT/ISSUE/REPORTS ON ORDS
             111Ø SECTORS REQUIRED ON ORDS (PK)
```

Sometimes simply putting a device online, mounting a tape, or loading a printer with paper are enough to reactivate a waiting process. Other times you must enter a system command to resume execution of the process. You can display the possible system command responses by entering the Y (Status Interrogate) command. For example, to see the responses for the first waiting entry in the preceding display, enter the mix number with the Y command, like this:

```
792Ø Y
```

Information similar to the following appears:

```
Status of Task 7916\7920 at 14:19:36
Program name: *LIBRARY/MAINTENANCE
Priority: 50
Origination: TA151/CANDE/1  (LSN 337)
MCS: SYSTEM/CANDE
Usercode: SUE
Chargecode: 6851
Stack State: Waiting on an event
RSVP: SUE/FILE000 REQUIRES MT #1
Reply: FA,OU,DS
```

The RSVP line shows the system message. The REPLY line shows the system commands you can enter in response to the RSVP message. Table 4–1 summarizes the possible system command replies to RSVP messages. You can use these commands to reactivate waiting entries only when the command actually appears in a REPLY line in the Y command output.

Table 4–1. Responses to Waiting Entries

Command and Command Name	Command Syntax and Example	Explanation
AX (Accept)	<mix number list> AX <your answer> 1234 AX WEEKLY	Lets you direct the program by answering an Accept message in the RSVP message (Accept messages are discussed in "Responding to a Prompt from a Waiting Process (AX)" in this section.)
DS (Discontinue)	<mix number list> DS 1234 DS	Causes the process to cease execution and terminate abnormally.
FA (File Attribute)	<mix number list> FA <attribute specification> 1234 FA FILENAME=OPS/START	Lets you assign a different file or library attribute from the one originally assigned to the process.
FM (Form Message)	<mix number list> FM <device> <unit number> 1234 FM LP20	Lets you restart a program that was suspended because it tried to open a file whose FORMID attribute is set.
FR (Final Reel)	<mix number list> FR 1234 FR	Indicates that the input tape reel just read by a tape unit is the final reel of an unlabeled tape file.

continued

Table 4–1. Responses to Waiting Entries (cont.)

Command and Command Name	Command Syntax and Example	Explanation
IL (Ignore Label)	<mix number list> IL <device> <unit number> 1234 IL MT32	Indicates that the input file requested by a process resides on a particular unit, regardless of the unit label.
NF (No File)	<mix number list> NF 1234 NF	Causes the processes specified to receive an open error result without necessarily being discontinued.
OF (Optional File)	<mix number list> OF 1234, 5678 OF	Treats files in a variety of ways according to the situation (see “Specifying That a File Is Optional (OF)” later in this section).
OK (Reactivate)	<mix number list> OK 1234 OK	Resumes a suspended process.
OU (Output Unit)	<mix number> OU <device> <unit number> 1234 OU MT28	Resumes a process and causes the output file being opened by the process to be written to a specified output device.
RM (Remove)	<mix number> RM 1234 RM	Resumes a process that was suspended because of a duplicate file condition and causes the system to remove the old file and keep the new file.
UL (Unlabeled)	<mix number list> UL <device> <unit number> 1234 UL MT20	Resumes a process suspended with a “NO FILE” message and assigns unlabeled files located on a particular unit to that unit.

The following topics describe the commands in Table 4–1 and the situations in which you would use them. If you need additional information or syntax for any command, see the *System Commands Reference Manual.*

Responding to a Prompt from a Waiting Process (AX)

Sometimes a programmer wants the operator to choose a particular course of action for a program, such as printing to a disk or a tape. The programmer tells the operator what choices to make for the program by writing an *Accept message* in the program. At a certain point in its execution, the program stops and enters the waiting entries with the Accept message. Accept messages usually look similar to this:

```
---Job-Task-Pri---Elapsed-----1 WAITING ENTRY--------------------------
  1528/1528  5Ø  :1Ø      JOB (SITE)TEST
   R: ACCEPT: SPECIFY REPORT TYPE. ENTER WEEKLY OR MONTHLY.
```

To respond to an Accept message, enter the mix number, the AX (Accept) command, and your response. The previous example shows a job waiting for a response. You can respond to the Accept message displayed by this job by entering

```
1528 AX WEEKLY
```

After you enter a response, the process resumes execution and appears in the active entries portion of your ADM screen.

If you do not want to accept either of the actions given to you in the accept statement, you can use the DS command to terminate the process. Using the DS command is discussed in Part 2.

There might be other actions you can take with an Accept message besides entering AX or DS. To see if there are other alternatives for a particular waiting process, use the Y command as described earlier in this section.

Terminating a Waiting Process (DS)

You can terminate a process by entering the DS (Discontinue) command. This command can be useful if a process is caught in a loop, is running out of sequence, or is in error, and these problems cannot be corrected by any other means. Before discontinuing a job or task, you can stop the process with the ST (Stop) system command (this command is described under "Jobs and Tasks" in Section 8). Then you can discuss the process with a programmer or system analyst to see if there is some other solution besides using the DS command. You should also be aware of any site policies regarding discontinuing processes.

The DS command is described in Section 8 under "Jobs and Tasks."

Note: *You cannot use the DS command on a task or program that has been programmatically locked with the LP (Lock Program) system command. The LP command prevents both the DS (Discontinue) and QT (Quit) system commands from terminating the specified tasks or programs.*

For more information about the LP, DS, and QT system commands, see the System Commands Reference Manual.

To terminate a locked program, you must enter the LP– (Unlock) system command, which removes the automatic lock condition for the code file.

```
<mix number> LP--
```

You can then use the DS or QT command to terminate the process.

If the process you discontinued initiated other jobs (was the "parent" process of other tasks), those "offspring" tasks are also terminated. You see a "PARENT PROCESS TERMINATED" error message each time the system terminates one of those tasks.

The DS command is also available as a primitive command, which you can use if the ODT is not responding to standard system commands. To terminate a task using the primitive DS command, enter

```
??<mix number> DS
```

> **Caution**
>
> Entering the primitive DS command without a mix number terminates all processes currently running on the system.

Changing the File Attributes Expected by a Waiting Process (FA)

You usually see the FA (File Attribute) command listed as a reply to a process that gets suspended with a "NO FILE" message, because some attribute, such as a file name, is different from the one declared in the program.

Sometimes another process creates the needed file, and the waiting process resumes automatically. However, you usually need to specify the correct attribute. Some of the file attributes you can specify values for are device kind, file name, family, file title, and serial number. You can also use the FA command to assign values to library attributes. The *I/O Subsystem Programming Guide* provides complete information about the file attributes that you can change. The *A Series System Software Utilities Operations Reference Manual* describes library attributes and their values. The following discussions explain how to use the FA command to specify values for some common file attributes.

You do not always need to enter the FA command to resolve a NO FILE condition. As an alternative, you can copy the needed file to the requested disk volume or mount a tape containing the requested tape file, and the process resumes automatically. As you become familiar with the processes that run on your system, you will be able to determine the response that is appropriate in each situation.

You can keep the system from putting a process in the waiting entries if you know what attributes the program needs before you execute it. You can specify the needed attributes at execution time by using file equations. See "Specifying Alternate File Attributes" in Section 3 for details.

Assigning a Device Kind with FA

Suppose a process appears in the waiting entries list with the following message:

```
---Job-Task-Pri--Elapsed------- 1 WAITING ENTRY ----------------------
   * 4237  5Ø    11:14         (HENRY)OBJECT/SYMBOL
            NO FILE ALGOL/TASK1 (CR)
```

The (CR) means that the process is looking for a file it assumes is a card reader file. (You can find a list of device abbreviations in Table 2–1.) ALGOL/TASK1 is the name of the card reader file that the process is trying to open.

If ALGOL/TASK1 is really a tape file, you can mount the tape that contains the file and then restart the process by entering

```
4237 FA KIND = TAPE
```

If ALGOL/TASK1 is a disk file, you can restart the process by entering

```
4237 FA KIND = DISK
```

The system searches for the file on the DISK family. If ALGOL/TASK1 is located on a family besides DISK, you must assign the family name rather than the device kind. Using the FA command to assign a family is discussed later in this section.

Assigning a File Name with FA

Suppose a process appears in the waiting entries with the following message:

```
---Job-Task-Pri--Elapsed------- 1 WAITING ENTRY ----------------------
   * 4237  5Ø    11:14         (HENRY)OBJECT/SYMBOL
            NO FILE ALGOL/DATA1 ON ACCOUNTING
```

Assume that the file name has been updated to ALGOL/DATA2. To reactivate this process, enter

```
4237 FA FILENAME = ALGOL/DATA2
```

Assigning a Family with FA

Assume that in the previous waiting entry the file name is correct, but the family name has been changed to SYSPACK. To reactivate this process enter

```
4237 FA FAMILY = SYSPACK
```

Assigning a File Title with FA

For the previous waiting entry, assume that both the file name and family name have changed. In this case, you would assign a new title by entering

```
4237 FA TITLE = ALGOL/DATA2 ON SYSPACK
```

Remember that the *file title* includes the file name plus the family name.

Assigning a Serial Number with FA

In the following waiting entry, the process could not find the tape named ALGOL that has a serial number of PROGR1.

```
---Job-Task-Pri--Elapsed------- 1 WAITING ENTRY -----------------------
    * 4237  5Ø    11:14        (HENRY)OBJECT/SYMBOL
        4237 NO FILE ALGOL/FILEØØØ (MT) [PROGR1] #1
```

To reactivate this process, you can override the serial number declared in the program and specify another serial number by using the FA command in the following format:

```
   4237 FA SERIALNO = "PROGR3"
```

This example directs the process to the tape that has the serial number PROGR3.

Note that if you get an RSVP message while using the COPY statement, you can use FA only to change the serial number. The same limitation applies to processes attempting to open the second or later reel of a multireel tape file. If you cannot resume execution of the process by changing the serial number, you should talk to your supervisor.

Assigning a Printer Form Request to a Printer (FM)

If your site prints directly to printers that use special forms, such as payroll checks or invoices, you might see an RSVP message that requests FM as a reply. FM is requested as a reply when a process expects a specific form to be loaded on a printer, but it cannot find a printer with that form. The RSVP message looks similar to the following:

```
--Job-Task-Pri---Elapsed----------- 1 WAITING ENTRY------------------
   * 2121  5Ø     1Ø:23      HOURLY/EMPLOYEES/WORKLOG
                REQUIRES FM:CHECKS, IP OR LP
```

In this example, the waiting process needs the form named CHECKS. IP and LP represent an image printer or a line printer, respectively.

To direct the sample waiting entry shown in the example to an unformed printer, you can enter the following command, which assumes that IP3 is an unformed image printer:

```
   2121 FM IP3
```

If the process needs to print to the specific form it is requesting, it is best to use the PS CONFIGURE command to assign a FORMID to a printer. Assigning a FORMID allows only the files that have the matching FORMID to print to the “formed” printer. The following steps explain how to assign a FORMID using the PS CONFIGURE command.

1. Enter the *PER LP* or *PER IP* command to see if the printer you want to use is busy. The printer message “SCRATCH” indicates that the printer is not currently printing.
2. Make the printer you want to use unavailable to the system by entering

```
SV <device> <unit number>; ex: SV LPT4
```

3. Load the printer with the forms you want to use.
4. Assign the needed form to the printer by entering the *PS CONFIGURE* command in the following format:

   ```
   PS CONFIGURE <device> <unit number> FORMID <form name>
   ```

   To assign the form named CHECKS needed in the previous example, you would enter

   ```
   PS CONFIGURE LP4 FORMID CHECKS
   ```

5. Make the printer ready to use by entering

   ```
   RY <device> <unit number>
   ```

   The waiting process resumes execution and begins to print.
6. After printing has finished, remove the form assignment from the printer by entering

   ```
   PS CONFIGURE <device> <unit number> - FORMID
   ```

   For example, to remove the form assigned previously in step 4, enter

   ```
   PS CONFIGURE LP4 - FORMID
   ```

If the FORMID requested by a process can be replaced by another FORMID, you can use the FA (File Attribute) command described earlier in this section to specify the correct FORMID. The format for the FA command is

```
<mix number> FA FORMID = <new form ID>
```

There must be a printer that has the new form ID, or else the process reappears in the waiting entries.

Specifying the Final Tape Reel (FR)

Unlabeled tape files have no start or end labels. When the system reads an unlabeled tape, it does not know when it has come to the last reel and therefore does not know when to stop reading. Thus, when the system reads to the end of an unlabeled tape (as indicated by a tape mark), it puts the process in the waiting entries list so that the operator can indicate whether any further reels exist. The process displays a message similar to the following:

```
---Job-Task-Pri-Elapsed------- 1 WAITING ENTRY -----------------------
    *  4242  5Ø   11:14          (HENRY)OBJECT/SYMBOL
                 NO FILE CUSTOMER/DATA (UNLABELED MT) #3
```

The number 3 at the end of the preceding RSVP message indicates that the process is waiting for the third reel of the tape file. If there are additional reels to be read for this file, enter the UL (Unlabeled) command described later in this section.

If there are no more tapes to be read, enter the following command to respond to the preceding example:

```
4242 FR
```

This command informs the process that it has reached the end of the file and causes the process to resume execution.

Directing a Waiting Process to a Different Device (IL)

You can use the IL (Ignore Label) command to indicate that the input file requested by a process resides on a different tape or disk drive from the one specified in the program. For example, assume that a file has been moved from the OPERATIONS family to a family named SYSPACK. A program that uses that file is set up to look for that file on OPERATIONS. When it cannot find the file, it displays the following RSVP message:

```
---Job-Task-Pri--Elapsed------- 1 WAITING ENTRY -----------------------
    *  6789  5Ø    11:14          (HENRY)OBJECT/DAILYCHECK
              NO FILE CHECK/SYSTIME ON OPERATIONS (PK)
```

This message indicates that the process was looking for the file named CHECK/SYSTIME on the OPERATIONS family. To direct the program to the SYSPACK family where the file is now stored, you would first find a unit number for a disk in the SYSPACK family by entering

```
PER PK
```

If the family has multiple disks, choose any disk in the family and use its unit number as in the following command:

```
6789 IL PK 16
```

Note that you cannot use the IL command to change the device kind (for example, disk to tape or tape to disk). You can only specify another device of the same kind.

If you want to change the device kind, you can use the FA command as long as it is included in the reply list. See the discussion of the FA command in Section 4 and the Y command in Section 8 for more details.

You can also use the IL command in response to “DUP FILE” messages. A “DUP FILE” message can occur when a process tries to open a tape file and two different tapes with the requested name are mounted. You can use the IL command to tell the process which tape drive to use.

Sending an Open Error Result to a Process (NF)

You can enter the NF (No File) system command in response to a “NO FILE” RSVP message. The process in the waiting entries receives an open error result when you enter the NF command. If the process is designed to handle such an error, the process can then continue running without having to be discontinued due to the NO FILE condition.

The NF command thus has an advantage over the DS (Discontinue) system command in that it enables the program to handle the error result if it can. If the process is not designed to handle the open error result, the NF command causes an open error that terminates the program.

It is a good idea to try the NF command, and the OF (Optional File) command on a "NO FILE" RSVP message before discontinuing the process.

Specifying that a File Is Optional (OF)

The OF (Optional File) command appears as a requested reply to a variety of RSVP messages. Following are the messages that generate this request. The paragraphs following each message discuss the effect of the OF command in each situation.

NO FILE

This message appears if a programmer has identified a certain file used by a program as an optional file. Usually the programmer tells you if there is an optional file. If a "NO FILE" message appears with a waiting entry, and you know the file is optional, you can enter the OF command. The program proceeds without the file.

DUP FILE

This message appears if a process is attempting either to replace a system disk file or to replace an old disk file with the same file title as that of the new file, and the operating system option AUTORM is reset. Entering the OF command causes that attempt to fail, thus retaining the old file. You should notify any person whose process receives this type of message. When AUTORM is set, the system removes the old file and retains the new file, unless the old file in question is a system file.

DUP LIBRARY

This message indicates that the system is attempting to create a new version of a file. If you enter the OF command, the system removes the new file and retains the old file. If you want to create the new file and remove the old file, enter the RM (Remove) command as described later in this section.

RECOPY REQD

This message appears when a copy process copying files encounters an error with a particular file. If you enter the OF command, the system skips over the file with the error and continues to copy the remaining files.

If the destination is a tape, the omitted file appears to be present when the copy has finished. You can list tape files by using the TDIR (Tape Directory) command described under "Tapes" in Section 8. The header indicates that the omitted file is present because the tape directory was written to the tape before the copy and ensuing error occurred. If a process tries to copy that file from the tape, the system displays a message indicating that the file was omitted.

REQUIRES PK <family name> <file name>

This message indicates that the process requires a disk family that is not available. Entering the OF command changes the REQUIRES PK condition to a NO FILE condition, giving you several new replies to use to solve the problem and reactivate the process.

Reactivating a Waiting Process (OK)

You can use the OK (Reactivate) command to resume execution of a waiting process.

Sometimes you use this command to reactivate a process that was stopped with the ST (Stop) command. In this situation, the RSVP message indicates that the program was stopped by the operator. You can also enter this command to restart certain processes that were stopped programmatically.

For the *RECOPY REQD* RSVP message, an OK reply means that library maintenance should try to copy the file again.

In some cases, OK appears in the REPLY list, but is not in itself enough to solve the problem that caused the process to be suspended. The following RSVP message shows a situation in which just entering OK is not enough to enable the process to complete execution:

```
---Job-Task-Pri--Elapsed------- 1 WAITING ENTRY ----------------------
    *  6789  5Ø    11:14         (HENRY)OBJECT/SYMBOL
                PK46 5Ø4 SECTORS REQUIRED ON DPMAST
```

This message indicates that there is not enough disk space for the process to use. Entering OK causes the process to resume; however, if the needed disk sectors are still not available, the process immediately stops again. To resolve this message, you can free disk space by using the SQUASH (Disk Allocation) system command or by removing files. (Sending a message to all users to remove unnecessary files also helps.) Both of these procedures are discussed in Part 2. When the needed sectors become available, the process resumes automatically.

Directing a Waiting Process to an Alternate Device (OU)

You can use the OU (Output Unit) command to override a program request to send output to a specific device. Entering OU for tape requests also overrides any tape serial number requested by the process. Following are some of the situations in which you can use the OU command.

If there is a particular serial number defined in a process, but there is no tape mounted with the requested serial number, the system puts the process in the waiting entries list. The RSVP message looks similar to the following:

```
---Job-Task-Pri-Elapsed------- 1 WAITING ENTRY ------------------------
    *  6789 50    11:14         (HENRY)OBJECT/SYMBOL
                TEST/PROG REQUIRES MT [TEST1] #1
```

This message indicates that the process (HENRY)OBJECT/SYMBOL requires a tape with a serial number of TEST1. You can use the OU command to override the requested serial number and direct the file to any ready tape unit that is write-enabled and contains a scratch tape volume (refer to "Purging Tapes (PG)," in Section 8.) Assuming that you want the process in the preceding example to write to the tape on MT 32, you enter

```
6789 OU MT 32
```

Note that you cannot use the OU command to direct output to a different device type, just to a different unit of the same device type.

Removing an Old Duplicate File (RM)

You can use the RM (Remove) system command for processes that are suspended with a duplicate library condition. A duplicate library message indicates that duplicate disk files exist. The RM command causes the system to remove the old file and retain the new file. A duplicate file condition can occur in any of the following cases:

- The process attempts to create a new file with the same name as that of an existing file.
- The process attempts to change the name of a file to the same name as that of another file.
- The process tries to copy a file to a destination where a file with the same name already exists.

The following RSVP message shows a duplicate library condition where the duplicate file is named OBJECT/RUNNIT:

```
7589 PK46 DUP LIBRARY OBJECT/RUNNIT ON DPMAST
```

To remove the old version of OBJECT/RUNNIT and retain the new version, enter

```
7589 RM
```

If you want to keep the old file and discard the new file, use the OF (Optional File) command described earlier in this section.

Note that you can prevent duplicate file conditions by using the OP (Options) command to set MCP option 5, AUTORM. When this option is set, the system automatically removes the old duplicate file, and the process proceeds normally. See "Setting and Resetting MCP Options (OP)" in Section 8 for instructions.

Directing a Waiting Process to an Unlabeled Tape (UL)

You can use the UL (Unlabeled) command to resume a process that was suspended with a NO FILE condition and to assign unlabeled files located on a particular unit to that process. Note that if the physical tape file is a labeled file, and you enter a UL command, then the file is treated as unlabeled (regardless of whether the physical file is created as labeled). In this case, the process reads the labels as the first data records.

If a process attempts to open an unlabeled tape file, an RSVP message such as the following appears:

```
9347 NO FILE CUSTOMER/DATA (UNLABELED MT) #1
```

If you want the process to use the unlabeled tape on tape unit 21, you would enter the following command:

```
9347 UL MT 21
```

If the process is accessing a tape file that is stored on multiple reels of tape, the process displays the preceding RSVP message each time it reads to the end of a tape. In the RSVP message, the process increments the last number by 1 each time. Thus the following example indicates that the process is waiting to read the third reel of an unlabeled multireel tape file:

```
9347 NO FILE CUSTOMER/DATA (UNLABELED MT) #3
```

If there is another reel to be read, you can respond with another UL command to direct the process to the next reel. If there are no more reels to be read, you can respond with the FR (Final Reel) system command described earlier in this section.

Section 5
Writing WFL Jobs That Execute Programs

If you regularly execute the same programs, you can save yourself extra typing by creating a WFL job to do the work for you. In addition to less typing, a WFL job provides several convenient advantages:

- A WFL job executes all the programs you put in it. You enter only one command to start the job.
- A WFL job can run when you are not present.
- A WFL job can restart itself automatically at a particular time each day.

This section provides instructions for using some basic CANDE commands and the basic WFL structure to write a simple WFL job. For details about using CANDE to write jobs and programs, refer to the *CANDE Configuration Reference Manual* and the *CANDE Operations Reference Manual*. For details about using WFL, refer to the *WFL Reference Manual*.

Creating a WFL Job

Two primary tasks are involved in creating a WFL job: writing the job and testing the job. To write and test a WFL job, you perform the following basic steps:

1. Access CANDE.
2. Make a job file.
3. Enter the WFL job statements.
4. Save the job file.
5. Issue a CANDE START <file name> SYNTAX command to test the syntax of the WFL job.

These steps are described in detail in the next two topics.

Writing a WFL Job

Assume that at the end of every 24-hour period, your site executes a program and a series of jobs that provide a summary of operations for the day. Instead of executing the program and the jobs individually, you can put them in a WFL job, and then execute the WFL job with a single command. The following steps explain how to write such a WFL job.

1. Access CANDE by doing the following:

   a. From the ODT enter *??MARC* to get into MARC mode. If the MARC log-on screen appears, skip to step c.

   b. Press the space bar and then press the XMIT key to display the log-on screen.

   c. Enter the log-on sequence to display the MARC home menu.

   d. From the Choice line enter *CA* to get into CANDE.

   The system transfers you to CANDE and initiates your CANDE session. Messages similar to the following appear:

   ```
   #A12:539 CANDE 38.121 AT MP021; YOU ARE TA154/CANDE/3(285)
   #SESSION 2773  17:44:06  08/10/92
   ```

2. Make a CANDE file. Decide what name to give the file that contains the WFL job, following the file naming conventions described in Section 1. Enter the file name with the following command:

   ```
   MAKE <file name> JOB
   ```

   The MAKE command creates a file. *JOB* specifies that the type of file is a job.

   Assume that you name the file DAILY/MAINTENANCE. After creating the file, the system displays a message similar to the following:

   ```
   #WORKFILE DAILY/MAINTENANCE: JOB
   ```

3. Enter the statements that make up the WFL job. You can enter START and RUN command statements in the same form in which you enter them individually. You can also include family substitution statements and file equations. Because CANDE requires line numbers, you must type a number at the beginning of each line as shown in the following example.

   Assume that you want to put three jobs and one program in your daily maintenance job. Your job might look similar to the following:

   ```
   1 BEGIN JOB DAILY/MAINTENANCE;
   2 START (OPERATOR1)CHECK/SPACE ON USERPACK;
   3 START USAGE/SUM ON MAINTPK;
   4 START PRINT/LOG ON MAINTPK;
   5 RUN *OBJECT/ACCOUNT/SUM ON SYSPACK;
   6 END JOB.
   ```

   The first line must contain the words BEGIN JOB followed by the name of the job. This can be the name you assigned in step 2 or another name of your choice. The job name appears in the mix display when the job is running.

   It is sometimes easier to track problems that occur with the job if the file name in step 2 is the same as the job name.

   Note: *The last line must contain the ending statement, END JOB.*

4. Save the file by entering the following CANDE command:

   ```
   SAVE
   ```

   The system displays a message informing you that your work source is saved.

5. To check the contents of the file, enter

```
LIST <file name>
```

   a. If you see any mistakes or anything you want to change, scroll the line you want to correct to the top of the screen.
   b. Then make the change, place the cursor on the last character of the line, and press XMIT.
   c. After making all your corrections, save the changes by entering *SAVE*.

Your WFL job is now written. Follow the instructions in the next topic to test the job for any syntax errors.

Testing a WFL Job

If your WFL job contains any syntax errors, it will not run. So, after writing and saving your job, you should test it and then make any necessary corrections. While still in the CANDE session, enter the following command to see if the syntax of your job is correct:

```
START <file name> SYNTAX
```

For the sample job from the previous topic, you would enter

```
START DAILY/MAINTENANCE SYNTAX
```

If there are no errors, the system displays messages similar to the following:

```
#RUNNING 3797
#
```

You can execute your job at any time with the START command as shown in the following topic, "Starting a WFL Job."

If there are syntax errors, the system displays error messages. Assume that in the preceding sample job, you forgot to put the ending semicolon (;) on line 3:

```
3 START USAGE/SUM ON MAINTPK
4 START PRINT/LOG ON MAINTPK;
```

When you test the syntax of the job, you get the following error message:

```
#RUNNING 3815
4 START PRINT/LOG ON MAINTPK;
        *
  ERROR: END OF STATEMENT EXPECTED
#SNTX
```

The second line of the preceding message shows how far the job progressed before being terminated by the syntax program. From this line and the "end of statement" message, you can surmise that a line before line 4 is missing the ending semicolon (;). At this point, you can correct the error by following these steps:

1. If you have been working on other files, enter the following command to get the file you want:

   ```
   G <file name>
   ```

2. Now list the contents of the file by entering

   ```
   P
   ```

   This enables you to page through the file to find the incorrect line.

3. Scroll the incorrect line to the top of the screen and make the correction.

4. To save the change, enter

   ```
   SAVE
   ```

After correcting the error, run the syntax program again. Once you are sure the job is correct, you can execute the job by using the START command as described in the following topic.

If you do not know how to return to ODT mode, refer to "ODT Mode, Transferring to (??ODT)" in Section 8.

Starting a WFL Job

When you start a job from the ODT, your START statement must include the usercode and family name that the WFL job is stored under. If you do not include a usercode and family name, the system looks for the job as a nonusercoded file on the DISK family. Assuming that the example file, DAILY/MAINTENANCE, from the previous topic is stored under the OPERATOR1 usercode on the SUPPORT family, the command to start the job is

```
START (OPERATOR1)DAILY/MAINTENANCE ON SUPPORT
```

Refer to Section 8 under "Jobs and Tasks" for more information on starting WFL jobs.

When you enter a START statement, the system does not begin executing the WFL job immediately. Instead, the system subjects the job to several intermediate steps, as follows:

1. The control card independent runner compiles the job using the WFLSUPPORT support library.

   (WFL differs from other programming languages in that WFL jobs are compiled each time they run.)

2. The system assigns a mix number to the WFL job and inserts it in a job queue.

   If you start a job in CANDE, CANDE displays the queue number it put your job in. If you start a job from the ODT, you must enter the SQ (Show Queues) command to see what queue your job is in.

   You can display the jobs in a specific queue by entering

```
SQ <queue number>
```

For example, entering *SQ 16* shows all the jobs in queue 16.

3. The system selects the WFL job from the job queue for execution. Then the WFL job appears in the Active Entries display.

Understanding Job Queues

A job queue is a waiting line for jobs submitted for processing. The purpose of a job queue is to enable the processor to work most efficiently by regulating the number of jobs that can run at one time. Job queues can also enforce fair use of system resources among processes.

By default, the system has a single job queue. (At least one queue must exist in order to run WFL jobs.) You might want to have additional job queues if you want jobs to have different treatment by the system. For example, you might want a queue for short jobs that gives them higher priority than long jobs.

You can use the MQ (Make or Modify Queue) command to create, modify, and remove job queues. There are a number of queue attributes that you can define for each queue with the MQ command. The most important attribute is MIXLIMIT. The MIXLIMIT limits the number of processes from a job queue that can run at the same time. The following subsection explains how to create a job queue and define a mix limit for that queue. You can get details about the other queue attributes by referring to the *System Commands Reference Manual.*

Creating a Job Queue

To create a job queue and set a mix limit for that queue, enter the MQ (Make or Modify Queue) command in the following format:

```
MQ <queue number> MIXLIMIT = <mix limit number>
```

The following text explains the variables in this command.

Queue Number

For the queue number, you can assign the next number in sequence (the default queue number is 0), or you can assign some number that is meaningful to your site. The queue number cannot exceed 1023, although you can have only 100 queues in existence at one time.

There are many ways you can manipulate job queues, including modifying attributes and deleting a queue. See “Job Queues” in Section 8 for details.

Mix Limit Number

You probably want to assign a relatively small number to the mix limit, depending upon how many jobs from that queue you think can run at one time without making excessive demands on the processor and memory.

For example, assume that you have decided that there should be only one long job in the mix at a time, but up to three quick jobs are acceptable. You can create a job queue with a MIXLIMIT of 1 for heavy jobs and a job queue with a MIXLIMIT of 3 for quick jobs.

To ensure that these programs run in the appropriate queues, you can include the PROCESSTIME attribute for the quick queue and assign a short processing time. For example, assume that you want only jobs that can complete in 60 seconds or less to run in the quick queue, so you assign a PROCESSTIME attribute value of 60 (seconds). Any job that took longer would run into the 60 second processor time limit and be discontinued. Users with long jobs would therefore be encouraged to run their job in the proper queue.

Now suppose someone starts a job and the system places that job in queue 5. Before initiating the job, the system compares the MIXLIMIT for the job queue with the active count for the job queue. (The active count for a job queue is the number of jobs in the mix that originated from that queue, plus one for each extra process running under those jobs. For example, a job that has one running process counts as one active entry; a job that has two processes running counts as two active entries; and so on.) If the active count is greater than or equal to the MIXLIMIT, the system does not initiate the new job. Later, when some of the jobs from that queue have completed, the active count is lower and the system initiates the new job.

You can use the ML (Mix Limit) command to display the current MIXLIMIT values and active counts of every job queue on the system. You might notice that the active count is higher than the MIXLIMIT in some cases. This situation can occur if a job runs two or more tasks simultaneously. The MIXLIMIT attribute does not prevent initiation of tasks, only initiation of jobs.

A WFL job itself can include a CLASS statement that specifies the number of the job queue for that job. If the job does not specify CLASS, then the system places the job in the default queue established by the DQ command.

The other queue attributes that limit resources function in a similar manner. IOTIME limits the I/O usage for a job and its tasks. LINES limits the number of lines printed. CARDS limits the number of CARDS punched. DISKLIMIT limits the number of disk segments allocated. PRIORITY limits the priority that can be assigned to the job or sets the priority if the job does not have one. If you have assigned your job to a queue, but the requirements for your job exceed the limits for that queue, the MCP discontinues your job.

Scheduling a Starting Time for a WFL Job

You can schedule a WFL job to start at or after a particular time by using the STARTTIME statement in one of the following three ways. Be sure to specify time in 24-hour format.

In the WFL Job Itself

You can include the STARTTIME statement near the start of the job in the group of statements referred to as the *job attribute list*. The following is an example of a job that includes a STARTTIME statement:

```
BEGIN JOB DAILY/MAINTENANCE;
STARTTIME = 11:ØØ;
  START CLEANUP/FILES ON MAINTDISK;
  START DAILY/UPDATE ON MAINTDISK;
  RUN *SETUP/CANDE/OPTIONS ON OPERATIONS;
END JOB
```

This job does not start until 11:00 a.m.

With the START Command

You can include a start time in the START statement that you enter to execute the job. Following is an example:

```
START DAILY/MAINTENANCE ON OPERATIONS; STARTTIME = 14:ØØ
```

This job does not start until 2:00 p.m.

In the Job Queue

You can change the starting time of a job in a queue. To do this, first display the mix number of the job by entering the SQ (Show Queue) command. Assume that the mix number for the job you want is 3475. The following command would assign a STARTTIME of 11:30 p.m. to that job:

```
3475 STARTTIME = 11:3Ø
```

In addition to structuring the STARTTIME statement to start a job at a particular time, you can also use STARTTIME in these formats:

STARTTIME = + 2:30	Do not start until two and a half hours from now.
STARTTIME = 23:15 ON + 1	Do not start until tomorrow at 11:15 p.m.
STARTTIME = 10:00 ON 11/03/92	Do not start until 10:00 a.m. on November 3, 1992.

Note: *There might be some delay between the STARTTIME and the time the job is actually initiated. For example, if the STARTTIME arrives and the active count for the job queue is greater than the MIXLIMIT, the job cannot be initiated right away, but starts as soon as possible thereafter.*

Occasionally a job cannot start processing even when it has queue space, because another running job has higher priority. When this happens, the job starts processing when an operator lowers the priority of the other job, or the other job finishes.

Writing a Job That Restarts Itself Daily

Sometimes there is a routine operations task that needs to be performed every day, and which can be written in WFL. A good example is library maintenance tasks, such as updating the versions of files on certain families. You can save yourself the trouble of having to start the WFL job each day. Because WFL includes a START statement, you can design the job to start itself over again. The following is an example:

```
BEGIN JOB DAILY/UPDATE;
CLASS = 2;       % Request job queue 2
  COPY (JONES)OBJECT/= FROM ENG179(PACK) TO USERDISK(PACK);
  START (OPERATOR1)DAILY/UPDATE ON OPERATIONS;
     STARTTIME = 7:ØØ ON + 1;
END JOB
```

This job is named DAILY/UPDATE and is stored in the file that has the name (OPERATOR1)DAILY/UPDATE ON OPERATIONS. The START statement at the end of the job therefore causes the same job to be executed again. The STARTTIME clause causes the next execution to be delayed until 7:00 a.m. on the following day. The job therefore runs automatically on a daily basis.

Writing a Job That Interacts with Operators

Suppose you write a WFL job that can be run by other operators. The operator might need some instructions to go with the job. For example, the operator might need to mount a certain tape that the job reads from. You can design the WFL job to display various messages at the ODT by using the DISPLAY statement. The following job uses a DISPLAY statement:

```
BEGIN JOB RUNNER;
  DISPLAY "MOUNT TODAY'S UPDATE DATA TAPE";
  RUN (SUSIE)OBJECT/UPDATE ON USERPK;
END JOB
```

The display message appears in the Messages list in the ADM display. However, the operator can easily overlook a display message, because the system displays so many other types of messages. To make absolutely sure that the operator sees the message before the job progresses any further, you can write the message as an ACCEPT statement. The following job shows an example of an ACCEPT statement:

```
BEGIN JOB RUNNER;
STRING INPUT;
  INPUT := ACCEPT("SPECIFY REPORT TYPE. ENTER: WEEKLY OR MONTHLY");
  RUN (SUSIE)OBJECT/UPDATE ON USERPK (INPUT);
END JOB
```

This job does several things. In WFL, the colon and the equals sign (:=) are the assignment operator, which in this case assigns a value to a string variable called INPUT. The value comes from an ACCEPT statement. When the job executes this assignment statement, it displays the message "SPECIFY REPORT TYPE. ENTER: WEEKLY OR MONTHLY" at the ODT. The job then becomes suspended and enters the Waiting Entries display, with "SPECIFY REPORT TYPE. ENTER: WEEKLY OR MONTHLY" as the RSVP message. The operator can respond with the AX (Accept) command. For example, if the job is running with mix number 1647, the operator can enter:

```
1647 AX WEEKLY
```

After this command, the job would resume execution and initiate the program OBJECT/UPDATE, passing the value WEEKLY as a parameter to the program.

Two other WFL features for communicating with operators are the INSTRUCTION statement and the FETCH specification. For details, refer to the *WFL Reference Manual.*

To avoid cluttering the ADM display area, it is a good idea to write only display messages that provide significant information, for example, messages that give the operator instructions or that show a program error or termination. Messages such as "This job completed successfully" take up unnecessary space and are redundant (the operator can see the job in the completed entries).

Section 6
Managing Disk Space, the System Log, and Printing

While most of your work involves monitoring and controlling the programs that run on your system, you also have to know how to manage disk files and printer operations. This section gives instructions for

- Checking available disk space
- Making backup copies of your disk directories
- Managing the system activity log
- Controlling printer operations

For information about other disk operations, see the topics "Devices" and "Disks" in Section 8.

For complete information on printing, see the *Print System Guide*.

Monitoring Available Disk Space

Available disk space is important for efficient system operation. The operating system must have enough available disk space to allocate areas for system and user files. See your system administrator for the policies regarding disk space maintenance at your site.

As an operator, you should be aware of the amount of available disk area on your system. The DU (Disk Utilization) system command displays information about the amount and size of the available areas on a given disk. This command is described under "Disks" in Section 8. Other commands related to maintaining available disk area are the SQUASH (Consolidate Disk Allocation) and the SCAN (Scan Disk or Pack Volume) system commands. These commands are also described under "Disks." Being familiar with the use of these commands can help you maintain open disk area and avoid disk space problems.

Making a Copy of the System Directory

The system uses the directory on the base disk of each family to locate the files stored on the disks in that family. If any portion of this directory becomes corrupted, you cannot retrieve all the data on the disks in that family. If you have multiple disk families, you should copy the system directory of each multidisk to another disk in that family to guard against losing the system directory, and thus losing data in the family. If the original directory gets corrupted, the system automatically uses the duplicate.

For instructions for copying the system directory, see "Disks" in Section 8.

You can make two duplicates of the system directory, for a total of three directories for each family. Three directories for each family is the maximum number of directories you can have. Note that the system updates the original and the duplicates automatically. This helps protect you from a corrupted system directory, but it uses system resources to do so. Talk to your system administrator about your site's policy in this area.

Maintaining System Activity Log Files

The system keeps a record of its activity, including operating system operations and process runs, in a file named SYSTEM/SUMLOG located on the family specified with the DL LOGS command. (The default location for this file is on the halt/load family.) The information in the SYSTEM/SUMLOG file is of vital importance in troubleshooting system problems and process terminations. To read the SYSTEM/SUMLOG file, you must use a program that can read the LOG file format. One such program, SYSTEM/LOGANALYZER, is described later in this section. While you might never need to analyze the SUMLOG file for problems (this is usually a job for system engineers or support personnel), you need to consider some maintenance issues, which are described in this section.

The active SYSTEM/SUMLOG file has a maximum size of 100,000 records. When the SYSTEM/SUMLOG file approaches 95 percent of the maximum size, the operating system changes the file name of the active log and creates a new log with the name SYSTEM/SUMLOG.

The operating system uses the following file name for naming old SUMLOG files:

```
SUMLOG/<system serial number>/<mmddyy>/<nnnnnn>
```

Each time the operating system changes the file name of the active SYSTEM/SUMLOG file to a new SUMLOG file, it increments the last node in the file name by 1. For example, assume that your system serial number is 1234, and the operating system has transferred the SUMLOG file three times during a week to new SUMLOG files, on July 11, 13, and 15 of 1992. The resulting SUMLOG files would be named as follows:

First transfer	SUMLOG/1234/07111992/000001
Second transfer	SUMLOG/1234/07131992/000002
Third transfer	SUMLOG/1234/07151992/000003

If you *cold start* (completely reinitialize) the system with the SYSTEM/ASD/LOADER program, the SUMLOG number returns to 000001 and the numbering starts over again.

It is important to note that you can run out of space on a disk if you let SUMLOG files accumulate. See "Copying System Log Files to Tape" later in this section for information on removing SUMLOG files from your disks.

Entering Comments into the System Activity Log

At times you might want to include some of your own comments about system conditions or operations in the SYSTEM/SUMLOG. For example, by mentioning that data comm is down at a particular time, you can help support personnel interpret system operations in the correct perspective. To include a comment in the SYSTEM/SUMLOG, you enter the LC (Log Comment) command in the following way:

```
LC <comment>
```

Your comment might look similar to this:

```
LC DATA COMM DOWN 07/11/92 FOR 2 HOURS STARTING 9:15 AM
```

If you are making comments about particular jobs, you can have those comments print out in the job summary by entering the comment with the LJ (Log to Job) command rather than LC. The format for LJ is very similar:

```
<mix number list> LJ <comment>
```

Your comment might look similar to this:

```
1234, 5678 LJ THESE ARE TEST PROGRAMS
```

Directing the System Log to a Family Other than the Halt/Load Family

The operating system puts SUMLOG files on the halt/load family by default. The halt/load family also contains all the system software (by default). To avoid running out of space on the halt/load family, you can direct the operating system to put SUMLOG files on another family. Eventually you should copy SUMLOG files to tape as described later in this section.

To direct SUMLOG files to another disk family, use the DL (Disk Location) command in the following format:

```
DL LOG ON <family name>
```

For example, if you want all the SUMLOG files to be written to the MAINTPK family, enter

```
DL LOG ON MAINTPK
```

Copying System Log Files to Tape

You should regularly copy SUMLOG files to tape and then remove them from the disk to conserve disk space. If you want a record of system activity on a daily basis, or if the SUMLOG files are accumulating quickly, you should copy the SUMLOG files to tape and remove them at the end of every work day. If the active SYSTEM/SUMLOG file is not full, but you want to copy what is there to tape anyway, enter the following command:

```
TL
```

This command changes the file name of the active log file and starts a new one, so you can copy and remove the old SUMLOG file.

To display SUMLOG files, enter

```
PD SUMLOG/= ON <family>
```

The family in the previous command is the family you assigned with the DL (Disk Location) command. You can issue a DL LOGS command to display this family name.

To copy all SUMLOG files to tape, enter the following command. Note that the tape serial number is optional. If included, it cannot exceed six characters.

```
COPY SUMLOG/= FROM <family>(KIND=DISK) TO <tape name>(KIND=TAPE,
SERIALNO="<serial number>")
```

Your command will look similar to the following:

```
COPY SUMLOG/= FROM MAINTPK(KIND=DISK) TO LOG1(KIND=TAPE, SERIALNO=
"SUM001")
```

Note that the copy might require more than one tape.

To copy a specific SUMLOG file to tape, enter the following command. Note that the tape serial number is optional. If included, it cannot exceed six characters.

```
COPY SUMLOG/<system serial number>/<mmddyy>/<number>
FROM <family>(KIND=DISK) TO <tape name>(KIND=TAPE,
 SERIALNO="<tape serial number>")
```

Your command will look similar to this:

```
COPY SUMLOG/1234/07111995/000001 FROM OPSPACK(KIND=DISK) TO LOG1(KIND=
TAPE, SERIALNO="SUM001")
```

You can find complete instructions for copying a file in “Files” in Section 8.

Printing the System Activity Log Files

To see the contents of the SUMLOG files, you must use a utility program that understands the LOG file format, such as SYSTEM/LOGANALYZER. This program analyzes the log and puts it in a form that you can print or display at the ODT. You can use the following commands to print the SUMLOG files. Refer to the *System Software Support Reference Manual* for the complete list of options you can use with the LOG command, for sample log analysis printouts, and for complete information on the LOGANALYZER utility. Table 6–1 lists the basic commands you can use to print system log files.

Table 6–1. Commands Used to Print the System Log Files

Task	Command
Print an analysis of a particular SUMLOG file	LOG "<sumlog file name>"
Print an analysis of a particular job	LOG JOB <mix number>
Print all the messages generated during a particular time span (for example from 6 p.m. to 6:30 p.m.) in chronological order	LOG 1800 TO 1830 MSG
Display on the ODT screen all log information for a particular job	LOG CON JOB <mix number> (CON stands for "console," which is the ODT.)
Print all the comments entered with the LC and LJ commands	LOG COMMENT
Print all mainframe errors, peripheral errors, and hardware configuration logs	LOG MAINT
Print a record of all DS commands entered	LOG OPERATOR DS
Print a record showing abnormal program terminations	LOG ABORT

Controlling Printer Operations

A special group of system commands, the PS (Print System) commands, controls the Print System. You can use a subset of those commands to control printer operations.

The *Print System Guide* discusses all the tasks for the Print System. If you are familiar with the basic tasks and just need to see the syntax of a PS command, you can refer to the *System Commands Operations Reference Manual.*

The most frequently used PS commands are introduced in Tables 6–2 through 6–6. The paragraphs following the table discuss the difference between direct and indirect (spooled) printing and explain how you can manage printer backup files.

Table 6–2. Commands That Display Information about Print Requests

Command	Description
PS SH	Shows all queued print requests
PS SH EXCEPTIONS	Shows all queued requests that have exceptions (errors) that prevent the request from being completed
PS SH WAITING	Shows all queued print requests that are waiting on a resource (such as an available printer)

Table 6–3. Commands That Display Information about Printers

Command	Description
PS DEV	Shows available printers and provides additional information, such as whether the printer is in the default pool (can be used by any process) or must be specifically requested by destination
PS DEV <device name>	Shows information for the specified device (Use PER LP to find the device names of your printers)

Table 6–4. Commands That Delete Print Requests

Command	Description
PS DELETE <request number>	Deletes the specified print request (use PS SH to get the print request number)
PS DELETE ALL	Deletes all print requests
PS DELETE EXCEPTIONS	Deletes all print requests with exceptions (errors)

Table 6–5. Commands That Force Print Requests to Print

Command	Description
PS FORCE <request number>	Forces the print request to print as soon as a printer is available
PS FORCE JOB <job number>	Same as PS FORCE

Table 6–6. Commands That Stop and Resume Print Requests

Command	Description
PS STOP <device name>	Immediately stops printing on the specified device
PS STOP <device name> AT EOF	Stops printing on the specified device when the current file has finished printing
PS OK <device name>	Resumes printing that was stopped by PS STOP

Using Direct or Indirect Printing

Processes running on your system can send output directly to a printer (direct printing) or to a special printer file on disk (indirect printing). With direct printing, a process is immediately linked with a printer, and no other process can use the printer until the original process has finished executing. There are times when the process has to wait for the printer to print output and times when the printer is idle while the process is executing. As you can see, this is not the most efficient way to use printer or processing time.

Indirect printing, however, sends print output to a disk file called a *printer backup file.* The printer backup files are sent to disk and are automatically printed when a printer becomes available. Thus, the program does not wait for the printer, and the printer does not wait for the program. For more information about direct and indirect printing, see "Print System" in Section 8.

You can enable direct printing in jobs and programs on an individual basis by setting the PRINTDISPOSITION attribute. You might want to use direct printing for controlled forms such as checks. Direct printing does not create a disk file, which ensures that privileged information such as check amounts cannot be accessed by unauthorized people. Direct printing is also advisable if you have to match program output with specific serial numbers on preprinted forms.

Understanding Printer Backup File Storage and Naming Conventions

With the LPBDONLY MCP option (option 4) set, when a program opens a printer file, the system automatically creates a printer backup file on disk. When a job finishes and a printer becomes available, the system prints and removes each backup file created by programs in that job. The system assigns each printer backup file a file name with the following form:

```
BD/ØØØ<job number>/ØØØ<task number>/<nnn><internal file name>
```

The last three digits before the internal file name are incremented by 1 each time a new file is created by that particular process. For example, assume that the following job generated three printer backup files:

```
---Job-Task-Time--Hist----------- COMPLETED ENTRIES ------------------
  5637\5652 Ø8:4Ø EOT   (HARVEY)OBJECT/SD ON TEST
```

Assuming that the internal file name is SDDATA, the printer backup files would be named as follows:

```
(HARVEY)BD/ØØØ5637/ØØØ5652/ØØ1SDDATA
(HARVEY)BD/ØØØ5637/ØØØ5652/ØØ2SDDATA
(HARVEY)BD/ØØØ5637/ØØØ5652/ØØ3SDDATA
```

Part 2
Operator Tasks

Section 7
Using Task Procedures

This part of the guide is a reference section containing step-by-step procedures for common operator tasks.

These tasks are alphabetized by subject—if you want to know how to clear a device, look up "Devices." Since you might not think of the same key word when looking for a task, the index in this book is important. If you cannot find a task by subject, try the index.

Each task has an explanation preceding the procedure. The explanation contains background information, reasons for using the task, and any issues related to the task. The procedure contains instructions on how to do the task.

Most task descriptions also include examples of task use.

Before you start a task, read the entire explanation and procedure to get a good idea of what is involved, so that you do not begin a task you cannot finish.

This section describes the most common operator tasks, and the tasks cover only the most basic methods. To keep things simple, this section commonly refers you to more detailed information in other books. It is therefore a good place to start looking for information.

Most of your work as an operator involves using system commands, which are frequently entered at the operator display terminal (ODT). This guide relies heavily upon information contained in the *System Commands Reference Manual.* It is a good idea to familiarize yourself with the reference manual, and to keep it close to the ODT for quick reference.

For easy command reference, Table 7–1 lists all the system commands described in this guide alphabetically, along with the subsection in which each command is described most thoroughly. Unless otherwise noted, all the tasks mentioned are in Section 8.

Table 7–1. System Commands in Operator Tasks

System Command	Subsection Task Is In
A (Active Mix Entries)	Mix
ACQUIRE (Acquire Resource)	Devices
AX (Accept)	Jobs and Tasks
C (Completed Mix Entries)	Mix

continued

Table 7–1. System Commands in Operator Tasks (cont.)

System Command	Subsection Task Is In
CL (Clear)	Devices
CLOSE (Close Pack)	Disks
CM (Change MCP)	master control program (MCP)
CU (Core Usage)	Memory Usage, Displaying (CU)
DBS (Data Base Stack Entries)	Data Base Stack Entries, Displaying (DBS)
DD (Directory Duplicate)	Disks
DIR (Directory)	Files
DL (Disk Location)	Print System
DR (Date Reset)	Time and Date
DS (Discontinue)	Jobs and Tasks
DU (Disk Utilization)	Disks
DUMP (Dump Memory)	Memory Dump, Requesting
FA (File Attribute)	Changing the File Attributes Expected by a Waiting Process (FA) (Section 4)
FM (Form Message)	Assigning a Printer Form Request to a Printer (FM) (Section 4)
FR (Final Reel)	Tapes
FREE (Free Resource)	Devices
FS (Force Schedule)	Jobs and Tasks
GC (Group Configuration)	System
HI (Cause EXCEPTIONEVENT)	Exception Event, Causing (HI)
HS (Hold Schedule)	Jobs and Tasks
ID (Initialize Data Comm)	Data Comm
IK (Install Keys)	Keys Files, Installing (IK)
IL (Ignore Label)	Directing a Waiting Process to a Different Device (IL) (Section 4)
J (Job and Task Structure Display)	Jobs and Tasks
LC (Log Comment)	Maintaining System Activity Log Files (Section 6)
LIBS (Library Task Entries)	Libraries, Displaying (LIBS)
LJ (Log to Job)	Maintaining System Activity Log Files (Section 6)

continued

Table 7–1. System Commands in Operator Tasks (cont.)

System Command	Subsection Task Is In
LOG (Analyze Log)	Maintaining System Activity Log Files (Section 6)
MCS (Display MCS Information)	Special Types of Processes (Section 3)
ML (Mix Limit)	Mix
MOVE (Move Job/Pack)	Job Queues
MP (Mark Program)	Messages, Displaying Identity of (MP)
MQ (Make or Modify Queue)	Job Queues
MSG (Display Messages)	Messages, Displaying (MSG)
MSC (Suppress Messages)	Messages, Suppressing (MSC)
MU (Make User)	USERDATAFILE, Modifying (SYSTEM/MAKEUSER, MU)
MX (Mix Entries)	Mix
NF (No File)	Sending an Open Error to a Process NF (Section 4)
NS (Next Screen)	Next Screen, Displaying (NS)
OF (Optional Files)	Specifying That a File Is Optional (OF) (Section 4)
OK (Reactivate)	Jobs and Tasks
OL (Display Label and Paths)	Devices
OP (Options)	master control program (MCP)
OU (Output Unit)	Devices
PD (Print Directory)	Files
PER (Peripheral Status)	Devices
PG (Purge)	Tapes
POWER (Power Up/Down)	System
PR (Priority)	Jobs and Tasks
PS (Print System) - All PS commands	Print System
QF (Queue Factors)	Job Queues
RB (Rebuild Access)	File Access Structure, Rebuilding (RB)
RC (Reconfigure Disk)	Disks
RM (Remove)	Removing an Old Duplicate File (RM) (Section 4)

continued

Table 7–1. System Commands in Operator Tasks (cont.)

System Command	Subsection Task Is In
RW (Rewind)	Tapes
RY (Ready)	Devices
S (Scheduled Mix Entries)	Mix
SB (Substitute Backup)	Print System
SC (System Configuration)	System
SCAN (Scan Disk or Pack Volume)	Disks
SHOWOPEN (Show Open Files)	Disks
SL (System Library)	Libraries
SM (Send to MCS or Data Base)	Sending Commands to an MCS or Database (SM)
SN (Serial Number)	Tapes
SQ (Show Queue)	Job Queues
SQUASH (Consolidate Disk Allocation)	Disks
SS (Send to Station)	Sending Commands to an MCS or Database (SM)
STARTTIME (Start Time)	Job Queues
ST (Stop)	Jobs and Tasks
SV (Save)	Devices
TDIR (Tape Directory)	Tapes
THAW (Thaw Frozen Library)	Special Types of Processes (Section 3)
TI (Times)	Jobs and Tasks
TL (Transfer Log)	Maintaining System Activity Log Files (Section 6)
TR (Time Reset)	Time and Date
U (Utilization)	System
UL (Unlabeled)	Directing a Waiting Process to an Unlabeled Tape (UL) (Section 4)
W (Waiting Mix Entries)	Mix
WM (What MCP)	master control program (MCP)
Y (Status Interrogate)	Jobs and Tasks

Section 8
Task Procedures

Database Stack Entries, Displaying (DBS)

Use the DBS (Database Stack Entries) system command to display all the active database stacks. A database stack is an area of memory containing all the information necessary to enable multiple users to access a database at the same time The DBS command enables you to see which databases are currently in use, and how many users are accessing each one.

Some databases are active most of the time, and their displays are suppressed, so that they do not take up space in displays. The following procedures explain how to display both suppressed and unsuppressed stack entries.

Procedure

To display all the unsuppressed database stacks, enter

```
DBS
```

To display all the active database stacks, enter

```
DBS ALL
```

Example

If you enter DBS, and there are active databases, you see something similar to the following:

```
---Mix-Pri-Usr------- 3 ACTIVE DATABASES ------------------------------
  8379  8Ø   1  STADB
  8329  8Ø   5  STBDB
  8335  5Ø   1  STCDB
```

From left to right, you see

- The mix number
- The priority
- The number of current users
- The database name

The heading line shows the number of active databases.

Data Comm

For an operator, the most important part of data comm is the DATACOMINFO file. This file contains a complete description of the data comm configuration. The MCP uses this file to initialize the data comm subsystem.

An operator occasionally needs to switch the DATACOMINFO files the system is running on, or to add or remove stations in the file.

You can add and remove stations on your system with the Interactive Datacomm Configurator (IDC) without bringing data comm down. You can find complete IDC documentation in the *IDC Operations Guide*. Note that if you do change data comm with IDC and data comm is running, the changes take effect immediately.

You might also need to bring the data comm system down or to start it up if it is not running. Procedures for bringing down and initializing data comm, and for switching DATACOMINFO files follow.

Initializing Data Comm

If data comm is not running for some reason, you can initialize it with a form of the ID (Initialize Data Comm) system command. If this command fails to start data comm, there might be a problem with your network support processor (NSP) or data communications data link processor (DCDLP). You might find an indication of what the problem is in the system messages.

Procedure

Do the following to initialize data comm:

1. Get the NSP or DCDLP number by entering

   ```
   PER DC-
   ```

   Look for the NSP entry. Note the number in the leftmost field.

2. Enter the following ID command:

   ```
   ID <NSP number>
   ```

   The NSP number is the number you found with the PER DC- command. If the NSP number is 108, for example, enter ID 108. This command initializes the specified NSP. You should see the following message:

   ```
   NSP INITIALIZING
   ```

Data comm should start after you use the preceding procedure. To verify that it did initialize, enter a PER DC command. If you see a job number attached to the NSP, data comm is running. In the following example, the job number 9419 is attached to NSP 110:

```
------------------------------- DC STATUS ------------------------------
  110 NSP (DC-DLP, LEVEL: 50.0)(9419)
```

If data comm did not initialize, no job number is attached to the NSP, and you see a completed entry for a task with the name NSP with a P-DS in the *Hist* column. If data comm did not initialize, you should contact your system administrator.

Note that if you have more than one NSP, you must initialize each NSP as described in the preceding procedure.

Note: *If your installation rns with MCP option AUTORecovery reset, you will have to restart data comm each time the system halt/loads.*

Switching DATACOMINFO Files

If you have modified the DATACOMINFO file to add or remove station addresses, you need to switch the DATACOMINFO file your system is currently running on with the new file. The following procedure describes how to do this.

This procedure interrupts data communications use for about 10 minutes.

You need to know the name of the DATACOMINFO file for this procedure. The default name for this file is SYSTEM/DATACOMINFO. The first node (SYSTEM) is called the *DC prefix*. When you use the ID (Initialize Data Comm) system command in the following procedure, use just the DC prefix, not the full name of the file (note that the DC prefix might be more than one node). The system automatically adds the /DATACOMINFO onto the prefix when you use this command. If you do use the full file name, you get an error.

To find out the DC prefix on your system, enter

```
ID
```

The DC prefix is listed on the first line of the display, after the *NIF:*. (NIF stands for Network Information File.) Use this prefix in the following procedure.

Before beginning the following procedure, you should notify the system users that data comm will be unavailable for approximately 10 minutes. You can use the SM (Send Message) system command, as described in "Sending Commands to an MCS or Database (SM)" in this section to notify users.

Procedure

To switch DATACOMINFO files, do the following:

1. Make a copy of the present DATACOMINFO file, in case you need it later. To copy the file, enter

   ```
   COPY <file name> AS <DC prefix>/SAVE/DATACOMINFO FROM <family>(PACK)
   TO <family>(PACK)
   ```

   The file name is the name of the present DATACOMINFO file. If you are using the default prefix, enter

   ```
   COPY SYSTEM/DATACOMINFO AS SYSTEM/SAVE/DATACOMINFO FROM <family>(PACK)
   TO <family>(PACK)
   ```

2. Change the name of the new file to the name of the system DATACOMINFO file (the DC prefix, followed by /DATACOMINFO). Use the following syntax:

   ```
   COPY <new file name> AS <DC file name> FROM <family>(PACK) TO
   <family>(PACK)
   ```

For example:

```
COPY NEW/DATACOMINFO AS SYSTEM/DATACOMINFO FROM
DATA(PACK) TO INFO(PACK)
```

3. Turn off MCP option 12 (AUTODC) by entering

   ```
   OP -12
   ```

   If this MCP option is on when you bring down data comm, the MCP restarts data comm as soon as the program has ended.

4. Stop any new jobs from entering the mix. Enter the HS (Hold Schedule) system command:

   ```
   HS
   ```

   You need to do this so that no jobs are trying to use data comm when you bring it down.

5. Identify the new file to the system by entering

   ```
   ID <DC prefix> ON <family>
   ```

   The DC prefix is the prefix name in the ID command display.

6. Bring data comm down by entering

   ```
   ID:QUIT
   ```

   To make sure the program has left the mix, look in the completed entries for the following:

   - SYSTEM/DATACOMSUPPORT
   - The NSP job (or jobs)
   - SYSTEM/COMS (if it was running)
   - SYSTEM/CANDE (if it was running)
   - Any other message control systems dependent on data comm

7. Look in the system libraries list to make sure no data comm libraries are still frozen. Enter

   ```
   LIBS ALL
   ```

   Look to see if any of the following libraries are still there:

   - SYSTEM/DATACOMSUPPORT.
   - Any support libraries for MCSs running on your system. If you have COMS running, for example, you would look for SYSTEM/COMS and COMS/ROUTER.

If any of these libraries are still there, you need to discontinue them with the DS (Discontinue) system command. Since they are usually locked, however, you must unlock the library first (you cannot discontinue a locked program). To unlock the library, enter

```
<mix number> LP-
```

Now you can discontinue that library by entering

```
<mix number> DS
```

Unlock and discontinue all the data comm libraries which are still there.

8. Turn on the MCP option AUTODC, which you turned off in step 1. Enter

   ```
   OP +12
   ```

   Set this option so that if data comm goes down for some reason on its own, the operating system restarts it automatically.

9. Initialize the NSP. To find out the NSP number, enter

   ```
   PER DC-
   ```

   Look for the NSP entry. Note the NSP number, and enter

   ```
   ID <NSP number>
   ```

   This command initializes the specified NSP and starts it.

10. Start job scheduling again by entering

    ```
    HS-
    ```

Data comm should restart. To verify that it did initialize, enter a PER DC command. If you see a job number attached to the NSP, data comm is running. In the following example, the job number 9419 is attached to NSP 108:

```
-------------------------------- DC STATUS -------------------------------
  108 NSP (BLOCKED, LEVEL: 15.78)(9419)
```

If data comm did not initialize, no job number is attached to the NSP, and you see the NSP in the completed entries with a P-DS in the *Hist* column. If data comm did not initialize, you should contact your system administrator.

You should now be running with the new DATACOMINFO file. You can verify the new DC prefix with the ID command. If there are small problems with the new file, you can usually fix them with IDC. If there are major problems, you can switch back to the DATACOMINFO file you were running on before. Since you copied the old file with the node *SAVE* following the DC prefix (step 1 in the procedure), you can change back to this file by entering the following command:

```
ID <DC prefix>/SAVE
```

If your DC prefix is SYSTEM, enter

```
ID SYSTEM/SAVE
```

You should then be running on the original DATACOMINFO file again. If not, contact your system administrator.

Taking Down Data Comm (ID QUIT)

You can use the QUIT option of the ID (Initialize Data Comm) system command to bring down data comm totally, or to bring down only a specified network support processor (NSP).

You might want to take data comm down if you are having problems with it, for example, if none of the terminals are communicating with the system, or if you see NSP error messages in the system messages. Sometimes you can fix these problems by taking down data comm and then bringing it back up again, rather than performing a halt/load on the system (halt/loads stop all processes on the system and can cause data recovery problems, so you should avoid them if possible).

If taking data comm down does not fix the problem, or you cannot get data comm to go down, you should talk to your system administrator about the situation. A halt/load is probably necessary to fix the problem.

If you have more than one NSP, and you notice NSP error messages for only one of the NSPs on your system, you can take just that one NSP down without affecting the rest of the data comm system.

If you have MCP option 12 (AUTODC) set, data comm comes up automatically after you take it down. If this MCP option is not set, you need to restart it yourself, as described in “Initializing Data Comm” in this subsection.

You can find out if this option is set by entering

```
OP 12
```

Procedures

- To bring data comm down, enter

  ```
  ID:QUIT
  ```

  Look in the completed entries for the NSP job. If you see this job, data comm is down. If you have more than one NSP, you should see a completed entry for each NSP that was running before you entered the QUIT command.

- To bring down a single NSP, enter

  ```
  ID <NSP number>:QUIT
  ```

  The NSP number is the number of the NSP you want to take down.

  This command does not affect other NSPs.

Devices

This subsection contains information concerning devices. A device is any piece of hardware used for input, output, or storage. A disk drive, a printer, a terminal, and the ODT are all examples of devices.

The tasks in this subsection apply to all devices. The operations described usually apply to any input, output, or storage hardware.

The System Configuration File

Your system has a file called the system configuration file. This file defines all the devices that your system can access. The system configuration file is named

```
<system name>/CONFIG  (example: AMP24B/CONFIG)
```

This file usually resides on the system halt/load unit. During a halt/load, the system accesses the system configuration file and acquires all the devices listed in it.

Occasionally, you need to make changes to the system configuration file (for example, when you have purchased two additional disk drives). Use either CANDE or the Editor to update the configuration file, and then perform a halt/load. Your system can then access the new devices.

Adding Devices to the System (ACQUIRE)

You do not need to change the system configuration file each time you want to free or acquire a device to use on another system. The ACQUIRE (Acquire Resource) system command adds resources to your current system configuration, but it does not change the system configuration file.

The main use of the ACQUIRE command involves multihost environments. If you have two or more systems, there might be some peripheral devices connected to more than one system. The system that can use the device is determined through the system configuration files.

If you want to switch device control from one system to another, use the FREE (Free Resource) system command to disconnect the device from the system that has control. Then use the ACQUIRE command on the other system to take control of the device. This way, the two systems do not attempt to share the same peripheral device.

The reconfiguration command ACQUIRE alters the current definition of the group. The results of these alterations are maintained across halt/load procedures unless the modifier :TEMPORARY is specified. If :TEMPORARY or :TEMP is specified, the device is acquired until the next halt/load, at which time it is freed.

You can still use the ACQUIRE command if you have only one system. If a peripheral device is not communicating with the system, try clearing and readying the peripheral device with the CL and RY commands. If that is ineffective, you can sometimes fix the problem by freeing the device, then acquiring it. This forces the operating system to establish contact with the device and check the device status, if possible.

Procedure

To add a device to the system configuration, use the following syntax:

```
ACQUIRE <device> <unit number>
```

Example

If you want to add tape drive 28 to your system, enter

```
ACQUIRE MT 28
```

The system responds with the following message:

```
MT 28 WILL BE ACQUIRED
```

Then, in a few seconds, the following shows up in the system messages:

```
MT 28 SUCCESSFULLY ACQUIRED
```

To acquire tape drive 28 to your system temporarily, enter

```
ACQUIRE MT 28:TEMP
```

After a successful acquire, tape drive 28 will remain in your group until the next halt/load.

The FREE command uses the same syntax:

```
FREE <device> <unit number>
```

Clearing a Device (CL)

The CL (CLear) system command clears all exception flags for a device, and might terminate tasks using that device.

An exception flag is an indicator that an error in data or in a process has occurred. An exception flag can prevent further use of a device because when the system encounters certain exception flags, it does not use the device. For example, if a tape drive is turned off during a copy operation, then turned back on, the system sets an exception flag on the drive. Even though everything looks normal, you cannot use the drive until you clear it.

If a device is cleared when it is in use, output to that device is lost.

Note that you cannot use the CL command to clear terminals. You must use the RY (Ready) command on terminals (see "Readying a Device (RY)" in this subsection).

If the device is owned by an MCS (such as CANDE), you can send the CL command to the MCS with the SM (Send Message) system command. See "Sending Commands to an MCS or Database (SM)" in this section.

Procedure

To clear a device, use the following syntax:

```
CL <device> <unit number>
```

Example

If you want to clear tape drive 28, you enter

```
CL MT 28
```

Remember that this command causes the process using the device at the time to be discontinued.

Directing Output to a Device (OU)

You can direct the output of a job to a specific device with the OU (Output Unit) system command. This command overrides locked units, serial numbers, and tape density specifications. You can use the OU command to respond to a variety of entries whose processes have stopped and are waiting for an RSVP entry.

You can direct output to any output device: MT, PK, CP, IP, and SC. Note, however, that the output must be formatted appropriately for the device you direct it to. For example, you should not display output formatted for a printer to an ODT because printer output has 132 characters on each line, and the ODT has only 80 characters on each line.

Procedure

Use the following general syntax to direct output to a device:

```
<mix number> OU <output device> <unit number>
```

Example

If you want to have the output from a program with the job number 8341 go to tape drive 118, you enter

```
8341 OU MT 118
```

Displaying Device Status (PER)

The PER (Peripheral Status) system command displays the status of peripheral devices, and tells you what devices are available to the system. You can abbreviate the command by entering just the letter P instead of the letters PER.

Use this command when you need the unit number of a device, or when you want to know what devices are connected to your system, if they are online, and what processes are using them.

This command works for any device type.

The OL (Display Labels and Paths) system command provides related information about label and path status for devices. This command is discussed in "Displaying Labels and Paths (OL)" in this subsection.

There are example PER command displays for disk units, tape units, and printers in the following examples. For more information about the PER command, and for explanations of all the PER command displays, see the *System Commands Reference Manual.*

Procedures

- To display status information about all the online peripheral units available to your system, enter

```
PER
```

- To use the command to display information about online devices of a particular type, enter

```
PER <device> or P <device>
```

- To see information about all devices of a device type, including unlabeled, not-ready, not-connected, and not-present devices, enter

```
PER <device>-  or P <device>-
```

Disk Unit Status Example

To see the status of all the online disks on your system, enter

```
PER PK
```

The system response looks similar to this:

```
-------------------------------- PK STATUS -----------------------------
44*B       [659Ø44] #1 DISK (18)
46*B       [659Ø46] #1 INVENTORY (2)
47*C       [659Ø47:659Ø46:46] #2 INVENTORY (1)
48*B       [659Ø48] #1 PAYROLL (11)
```

For each disk you see (from left to right):

- The unit number.
- The write-enable indicator: asterisk (*) if enabled, blank if disabled.
- The letter B, C or U, for Base, Continuation, or Offline disk.
- The serial number, enclosed in square brackets. If the disk is a continuation disk, you see the serial number, the serial number of the base disk, and the unit number of the base disk, separated by colons (:).
- The family index.
- The family name.
- The number of open (in use) files on the disk, enclosed in parentheses.

Disk units that display the letter U after the unit number are not available for normal use. A message follows the U, indicating why the disk unit is unavailable. For example, the following message shows that disk unit 99 is not available:

```
99*U NOT READY
```

Some other messages that might appear follow:

Message	Explanation
UNLABELED	An error occurred when the system tried to prepare the disk for use on the system. Discuss the message with your system administrator.
BLASTED	The system has encountered a catastrophic error on the disk that it cannot recover from. You can try to recover the disk by saving it with the SV (Save) command, closing it with the CLOSE command, and then readying it with the RY (Ready) command. If this procedure does not work, a halt/load will work in some cases. If you try both of these options and the “BLASTED” message remains, consult your system administrator. The “Disks” subsection tells you how to save and close a disk, and “Halt/Load, Performing (??PHL)” tells you how to perform a halt/load.

continued

continued

Message	Explanation
SAVED	The disk has been made logically unavailable to the system with the SV (Save) command. When the SV (Save) command is entered for a disk, the system marks it as saved but enables open files to continue to be used. No new files can be opened. Usually, disks are saved when they need to be taken offline for some reason, but you must wait for all files to close. You can make a saved device available to the system again by entering the RY (Ready) command.
NOT READY	The disk is not available to the system because it is either logically or physically disconnected. Make sure that the disk is not powered off or physically disconnected. You can make a disk logically available to the system by entering the RY (Ready) command in the following format: `RY PK<unit number>`
RESERVED	The disk was made unavailable to the system with the UR (Unit Reserved) command. When a disk unit needs maintenance, this command is used to make the disk unavailable to the system. To make the disk available again, enter `UR- PK <unit number>`

Tape Drive Status Example

The following command displays the status of your tape drives:

```
PER MT
```

The system displays information similar to the following:

```
-------------------------------- MT STATUS ----------------------------
    115*P   [MYTAPE]  1600  #1  <12/07/90> MASTER/TEST
    116*P   [TAPE12]  (2321)  1600  #1  <8/30/90> CAN/FILE000
```

You see, from left to right:

- The unit number
- The write enable status: an asterisk (*) if the tape has a write ring, a blank if it does not
- The type of tape:

P	Phase encoded (PE) or group-coded recording (GCR) tape
Q	Quarter-inch cartridge tape
H	Half-inch cartridge tape
9	9-track tape

- The serial number

- If the tape is in use, the mix number of the process using the tape, in parentheses
- The tape recording density
- The reel number
- The cycle and version
- The creation date of the tape
- The tape name

In place of the serial number and the usual information about the tape file, you might see one of the following messages:

Message	Explanation
SCRATCH	A *scratch* tape is mounted on that tape drive. A scratch tape is a tape that has been purged of all files and is ready to have files written to it.
UNLABELED	The tape mounted on this tape drive does not have a label. A label shows what files are on the tape and displays other characteristics about the tape. Before a program writes to an unlabeled tape, it sends you a message so that you can confirm that the program should write to an unlabeled tape. See Section 4, "Responding to Waiting Processes," for information on responding to "UNLABELED" messages.
NOT READY	The tape drive is not available to the system. The tape device might not be online. Ready it by pressing the tape device ONLINE button. You might be also able to make it available or "ready" by entering `RY MT<unit number>`
SAVED	The tape drive has been made unavailable to the system with the SV (Save) command. You can make a saved device available to the system again by entering the RY (Ready) command.
RESERVED	The tape drive was made unavailable to the system with the UR (Unit Reserved) command. When a tape drive needs maintenance, this command is used to make the tape drive unavailable to the system. To make the tape drive available again, enter `UR- MT <unit number>`

Printer Status Example

Enter the following command to see the status displays for printers:

```
PER LP
```

The system response is similar to the following:

```
------------------------------ LP STATUS ------------------------------
    4    S C R A T C H EBCDIC96
    5 (6364)    SERVER/LP5/"R#8168"/"J#6168" EBCDIC96
```

From left to right, you see:

- The unit number.
- If the printer is in use, the mix number of the print server or process using the printer. *MCP* might also be displayed in this field if the operating system has control of the printer.
- The printer status. The printer status modes are described at the end of this list.
- If the printer is in use, the name of the print server or program using the printer. If the printer is not in use, the word SCRATCH appears.
- The value of the TRAINID attribute presently assigned to the printer. This describes the character set available to the printer.

The status modes you might see are as follows:

Message	Explanation
FORMED	The printer has a print form assigned to it so that it will print in a special format. The form name follows the word FORMED.
SAVED	The printer has been saved with a SV (Save) system command.
SCRATCH	The printer is idle and ready to print. If the printer has been designated as the printer where all output is automatically directed, the words DEFAULT DESTINATION appear in parentheses.
RESERVED	The printer has been reserved with a UR (Unit Reserved) system command.
NOT READY	The printer is not physically ready.

Displaying Labels and Paths (OL)

You can display label and path information for peripheral devices with the OL (Display Label and Paths) system command.

A *path* is a set of hardware addresses that uniquely describes the data flow between the host and a peripheral device.

The OL command tells you if the path to a device is ready, and whether the device firmware, if any, is loaded. This information is valuable if, for instance, a disk is not communicating with the system, but looks normal in the PER PK command display. enter an OL command for the disk to see if there is something wrong with the path or firmware. If the firmware shows "NOT LOADED" in the OL display, you can try readying the device with the RY (Ready) system command. This command is described in "Readying a Device (RY)" in this subsection.

If you have a disk drive that requires controlware, and the controlware is not loaded, use the LH (Load Host) system command to load the proper controlware. The LH command is discussed in the *System Commands Reference Manual.*

The PER (Peripheral Status) system command provides related information about device status. This command is discussed in this subsection under "Displaying Device Status (PER)."

For an explanation of the OL command displays, see the *System Commands Reference Manual.*

Procedure

To display label and path information, use this general syntax:

```
OL <device> <unit number>
```

Example

If you want to display the label and path information for disk unit 44, you enter

```
OL PK 44
```

The system would display something similar to the following:

```
PK    44 B       [659Ø44] #1 SYS39Ø (Ø)
CREATED ON: Ø1/23/92 AT 2Ø:55:Ø4
CREATION SITE 5Ø3
B9387S/659 (INTERLACED)
CAPACITY: 3Ø128ØØ SECTORS (5423Ø4ØØØ BYTES)
DLP       CONTROLWARE STATUS
11Ø6           UC     ONLINE
```

Readying a Device (RY)

Use the RY (Ready) system command to ready devices. The RY command makes devices ready for system use if they are inaccessible because they were

- Saved with a SV (Save) command
- Closed with a CLOSE (Close Pack) command
- Locked by a program
- Locked with an RW (Rewind) command
- Locked with an SNL (Serial Number) command
- Locked with a PGL (Purge) command

You can use the PER (Peripheral Status) system command to determine the status of a device. See "Displaying Device Status (PER)" in this subsection for an explanation of the PER command.

If the device communicates through data comm (the device is owned by a message control system) you can send the RY command to it with the SM (Send Message) system command. See "Sending Commands to an MCS or Database (SM)" in this section.

Procedure

To ready a device, use the following syntax:

```
RY <device> <unit number>
```

Example

If you want to ready tape drive 113, enter

```
RY MT 113
```

The system responds with a message similar to this:

```
MT 113 WILL BE READY
```

And then the following in the system messages:

```
MT 113 READY
```

Releasing Devices from the System (FREE)

The FREE (Free Resource) system command removes resources from the current system configuration.

The main use of the FREE command involves multihost environments. Some peripheral devices might be connected to more than one system if you have two or more systems. The system that can use the device is determined through the system configuration file.

If you want to switch device control from one system to another, use the FREE command to disconnect the device from the system that has control. Then use the ACQUIRE (Acquire Resource) command on the other system to take control of the device. This way, two or more systems do not attempt to share the same peripheral device.

You can, however, still use the FREE command if you have only one system. If a peripheral is not communicating with the system, try clearing and readying the peripheral. If this is ineffective, you can sometimes fix the problem by freeing the device, then acquiring it. This forces the operating system to establish contact with the device and check the device status, if possible.

Procedure

To free a device, use the following syntax:

```
FREE <device> <unit number>
```

Example

If you want to release tape drive 28 from your system, you enter

```
FREE MT 28
```

The system responds with the following message:

```
MT28 WILL BE FREED
```

Then, in a few seconds, the following shows up in the system messages:

```
MT 28 SUCCESSFULLY FREED
```

The ACQUIRE command uses the same syntax:

```
ACQUIRE <device> <unit number>
```

Making Devices Inaccessible to the System (SV)

You can make a device inaccessible to the system with the SV (Save) system command.

Use the SV command anytime you want to prevent the system from using a device. If, for example, you want to keep the system from using a printer, you can save the printer so that the system does not queue print requests to it.

The SV command does not interrupt a job that is using the device when the command is entered, but it prevents any new files from being assigned to it.

To make the device available to the system again, use the RY (Ready) system command.

Procedure

To save a device, enter

```
SV <device> <unit number>
```

Example

If you want to make line printer 4 inaccessible to the system, you enter

```
SV LP 4
```

The system responds with a message similar to this:

```
LP 4 SAVED
```

Disks

Disks are the online, random-access storage devices for your system. A 1 through A 6 systems usually have between one and four in-cabinet disk drives, often referred to as disk units. Larger systems have separate disk units containing fixed or removable disks.

Closing Disk Units (CLOSE)

The CLOSE (Close Pack) system command logically detaches a disk unit from the system and releases the label information for that unit.

You can use the CLOSE command to prevent the system from using a disk unit. You can use the UR (Unit Reserved) or the FREE (Free Resource) system commands on a closed disk unit.

You cannot close a unit if its disk has any files open. If it does, the command aborts with a "UNIT IS IN USE" message. You can use the SHOWOPEN (Show Open Files) system command to identify the open files and the tasks using them. A pack can be forcibly closed by specifying ":DS" with the command, as long as none of the open files are critical to the system.

You can use the RY PK <unit number> command to reestablish the disk label information of a closed disk unit.

For information about the PER PK and RY PK commands, see the entries under "Devices" in this section.

Procedures

- To close a disk unit, enter

  ```
  CLOSE PK <unit number>
  ```

  The unit number is the number of the unit you wish to close. You can find out the unit number of a disk with the PER PK command.

- To forcibly close a disk unit, enter

  ```
  CLOSE PK <unit number> :DS
  ```

Example

If you want to close unit number 64, you enter

```
CLOSE PK 64
```

The system responds first with the following message:

```
PKØ64 WILL BE CLOSED
```

Then, the following appears in the system messages:

```
PKØ64 UNIT CLOSED
```

You can verify that the unit is closed by entering the PER PK command.

Consolidating Disk Space (SQUASH)

When disk areas are allocated for tasks and then freed up again, the disk tends to become *checkerboarded*—that is, most of the available space on the disk ends up in small areas spread around the disk. The SQUASH (Consolidate Disk Allocation) system command attempts to move the files stored on a disk to one or two continuous areas on the disk.

The SQUASH operation increases the amount of usable free area on a disk by consolidating the small free areas into larger continuous free areas. This consolidation makes it easier for the operating system to allocate disk space.

When the operating system allocates area to store a file, it needs one or more contiguous areas at least as big as the areasize of the file. If all the free areas on the disk are in small chunks, the operating system might not be able to find a large enough contiguous available area in which to place an area of a new file. Using the SQUASH command helps the operating system by collecting most of the available disk space in one place.

For efficient disk space management, it is important to routinely use the SQUASH command on every disk unit. Your site should have some provisions for this, perhaps including a SQUASH in the set of daily routine jobs run on your system. Ask your system administrator about your site's policies.

Use the SQUASH command if you notice that a large amount of the available segments are in blocks less than 504 sectors long. Use the DU (Disk Usage) system command to display information about the available areas on a disk. The DU command is described in "Displaying Available Sectors (DU)" in this subsection.

The SQUASH command can also be used in response to a "SECTORS REQUIRED" message. If you see such a message, be sure to notify your system administrator.

The SQUASH command works best when there are few open files on the disk.

Only one SQUASH command can be active on the system at one time.

Procedures

- To consolidate the available area on a family, enter

  ```
  SQUASH <family name>
  ```

  If the family has more than one disk, the process consolidates the available area on each family member. Each disk is consolidated separately. No files are moved between family members.

- To consolidate the available area on a single family member, enter

  ```
  SQUASH <family name> (<family index>)
  ```

  Only the family disk specified with the family index is processed with the SQUASH operation. You can use the PER (Peripheral Status) system command to get family index numbers. This command is described under "Devices" in this section.

Creating Duplicate Directories (DD)

The DD (Directory Duplicate) system command creates or removes a backup copy of a family system directory (also referred to as *flat directory*).

For an explanation of the function of the system directory, see “Disk Organization” in Section 1.

It is a good idea to have a backup copy of the system directory for a family on one or more of the continuation disks. If the original system directory becomes corrupted, you can use the backup copy to restore it. If you do not have a backup copy, you might lose access to many or all of the files on a disk family.

The system names the duplicate directory SYSTEMDIRECTORY/nnn, where nnn is the family index of the disk receiving the duplicate.

After creating the new duplicate, the system automatically keeps it up to date. Every change in the original directory (for example, when a new file is added to the family) is also made in the duplicate directory.

Note that creating duplicate directories helps protect against information loss, but requires pack space and resources to keep the duplicate up to date.

Procedure

To create a duplicate system directory for a family, enter

```
DD ON <family name> (<family index>)
```

The family name is the family you want the duplicate directory for. The family index specifies which disk in the family you want the duplicate on.

Example

If you want to create a duplicate system directory for the family INVENTORY, and you want the copy to be on the continuation disk with the family index of 2, you enter

```
DD ON INVENTORY (2)
```

Displaying Available Sectors (DU)

The DU (Disk Utilization) system command displays the total number of available segments and the size, in segments, of the largest available area on a family.

A segment is the smallest addressable area on a disk, and is 30 words (180 bytes) long.

Use the DU command to monitor the amount of available space on your disks. You need to do this so that you can maintain enough open disk space for efficient space allocation. Using the SQUASH (Consolidate Disk Allocation) system command frequently is important for preventing this sort of disk space problem. This command is discussed in "Consolidating Disk Space (SQUASH)" in this subsection.

Ask your system administrator to determine the amount of available space you need to maintain on your system. Then, when you notice a family reaching that limit, copy all files that are not presently needed to tape; use the *ARCHIVE ROLLOUT* WFL statement to remove those files from the disk.

Procedures

- To display the available segments on a family, enter

```
DU ON <family name>
```

- To display the available segments on a single family member, enter

```
DU ON <family name> (<family index>)
```

Example

If you want to display the amount of available segments on the second disk in the family PAYROLL, you enter

```
DU ON PAYROLL (2)
```

The resulting display looks like this:

```
AVAILABLE=1403819, LARGEST AREA=589653
126333 SECTORS IN 1630 AREAS LESS THAN 504
1277486 SECTORS IN 30 AREAS LARGER THAN 503
2521 AREAS OF 504 SECTORS COULD BE ALLOCATED

CAPACITY OF FAMILY: 4829370 SECTORS (869286600 BYTES)
```

Displaying Open Disk Files (SHOWOPEN)

The SHOWOPEN (Show Open Files) system command reports the open files and the mix numbers and names of the processes using those files on a disk or disk family. This report is displayed at the ODT.

An open file in this context is any in-use area of a disk, and includes files such as job files. Each non-system file in the report is followed by the mix numbers and names of the file users. Operating system files, such as the MCP code file, CATALOG, LOG, and USERDATA, are shown in the report, but are not followed by a list of users.

If you invoke the SHOWOPEN command on a member of the halt/load family or overlay family, the report header indicates that the disk is a member of such a family.

Procedures

- To get a report of the open files on a single disk, enter one of the following commands:

  ```
  SHOWOPEN ON <family name> (<family index>)
  SHOWOPEN PK <unit number>
  ```

  The two preceding commands are synonymous.

- To get a report of the open files on a disk family, enter the following command:

  ```
  SHOWOPEN ON <family name>
  ```

Examples

If you want a report of the currently open files on the family DISK, you enter

```
SHOWOPEN ON DISK
```

The resulting display looks like this:

```
                  DISK IS THE HALT/LOAD FAMILY
(UNISYS)SYSTEM/MCP/39SR0 (MCPCODEFILE) System File
*SYSTEM/ACCESS/001 (CATALOG) System File
*SYSTEM/SLOG (DATA) System File
*SYSTEM/SUMLOG (DATA) System File
*SYSTEM/USERDATAFILE (DATA) System File
*JOBDESC (JOBDESCFILE) System File
*SYSTEM/WFLSUPPORT (DCALGOLCODE) SLed
    5442/5442 *SYSTEM/WFLSUPPORT
Job File for 5442
*SYSTEM/GENERALSUPPORT (NEWPCODE) SLed
    5443/5443 *SYSTEM/GENERALSUPPORT
Job File for 5443
*SYSTEM/PRINT/SUPPORT (NEWPCODE) SLed
    5444/5444 *SYSTEM/PRINT/SUPPORT
*SYSTEM/BACKUPFILELIST (DATA)
    5444/5444 *SYSTEM/PRINT/SUPPORT
*SYSTEM/PRINTERINFO (DATA)
    5444/5444 *SYSTEM/PRINT/SUPPORT
*SYSTEM/PRINT/ROUTER (DCALGOLCODE)
    5448/5448 *SYSTEM/PRINT/ROUTER
Job File for 5448
```

Reconfiguring Disks (RC)

The RC (Reconfigure Disk) system command purges all files from a disk and creates a new set of volume labels on that disk.

If you use this command, you lose all the information on the disk. Make sure you do not need anything on a disk before you reconfigure it.

The RC command is useful when you want to create a disk family or add a continuation disk to an existing family. If you want only to rename a disk, but do not want to purge it as well, you can use the LB (Relabel Pack) system command.

The RC command creates a new set of volume labels on a disk. A volume label is an area on a disk where the family name and serial number are stored. If a disk is a member of a multiple disk family, either a base disk or a continuation disk, then the name and serial number reflect that relationship. Because of this, you need to know if a disk is associated with other disks before you reconfigure it.

It is easy to lose information when reconfiguring a base disk in a multimember family, or when reconfiguring a continuation pack. Before reconfiguring a disk in a multimember family, you should read the description of the RC command in the *System Commands Reference Manual* for more information on reconfiguring family disks.

For an explanation of how the disks in a family interact, see "Disk Organization" in Section 1.

You can find out if any of the online disks on your system are members of a family with the PER (Peripheral Status) system command. The PER command shows you the disk name, serial number, and family index, among other things.

For example, if you enter PER PK on your system, you see something similar to this:

```
------------------------------- PK STATUS ----------------------------
44*B       [659044] #1 DISK (18)
46*B       [659046] #1 INVENTORY (2)
47*C       [659047:659046:46] #2 INVENTORY (1)
48*B       [659048] #1 PAYROLL (11)
```

If a disk is a base disk, it has a B next to the disk number (the number at the left of the display). If it is a continuation disk, it has a C next to the disk number, and a family index greater than 1. In the preceding example, disk 47 is a continuation disk in the family INVENTORY, whose base pack is disk 46.

Note that some base disks have no continuation disks. They are single-disk families.

Before you reconfigure your disk, perform the PER PK command. Note the pack number, the serial number, the family index, and the name of your disk. You need this information to perform the RC command.

Following are procedures for reconfiguring a disk that is not a member of a disk family, and for adding a continuation disk to a family.

A disk must be ready, write-enabled, and not have any files open when you enter the RC command. The disk might have been saved with an SV (Save) command, but must not have been reserved with a UR (Unit Reserved) command.

Note: *The RC command erases all information presently on a disk. Make sure you do not need any information on a disk you are planning to reconfigure.*

Procedure for Reconfiguring Non-Family Disks

To reconfigure a disk using the disk's present serial number and family name, use the following syntax:

```
RC PK <unit number> OLDNAME = <family name>
```

The family name is the present name of the disk. If you do not include this in the command, a waiting entry appears asking you to confirm the family name.

To reconfigure a disk with a new serial number or a new family name, or both, use the following syntax:

```
RC PK <unit number> NAME = <family name> OLDNAME = <family name>
SERIAL = <number>
```

If you want only a new family name or only a new serial number, you can leave NAME or SERIAL out of the command. If you leave OLDNAME out, a waiting entry appears asking for the present name of the disk.

You cannot assign the names TAPE or DISKPACK to a disk. The serial number must be in the range 1 through 999999.

Procedure for Reconfiguring Continuation Disks

Reconfiguring a continuation disk is the same as reconfiguring a disk, except that you must identify the continuation disk to its base disk so that the base disk can include it in the base disk directory.

Use the following syntax to reconfigure a continuation disk:

```
RC PK <unit number> NAME = <family name> OLDNAME = <family name>
SERIAL = <number> BP = <serial number>
```

In the preceding syntax,

- The unit number is the unit number of the disk you are reconfiguring.
- The NAME is the family name of the reconfigured disk's new base disk (since they will be in the same family, they have the same family name).
- The OLDNAME is the disk's old family name.
- SERIAL is the disk's serial number.
- BP is the base disk's serial number.

Example

Suppose you have two disks on your system: DISK2 and ORDERS. You want to reconfigure DISK2 as a continuation disk of the disk ORDERS.

Several days before the reconfiguration, your system administrator should ensure that all users of DISK2 have been warned of the change, and have had their files and family statements transferred to other packs. Just before the planned reconfiguration of DISK2, back up all files on the pack to tape. This is to prevent inadvertently losing any user files.

After the backup finishes, perform a PER PK command:

```
-------------------------------- PK STATUS -----------------------------
44*B        [659Ø44] #1 ORDERS (14)
46*B        [659Ø46] #1 DISK2 (4)
```

DISK2 still has files open, shown by the number in parentheses after the disk name. Send out a warning to all users that the DISK2 pack is going to be reconfigured. Then save the disk:

```
SV PK46
```

You wait a few minutes for the files to close. When there are no files open, you enter the RC command:

```
RC PK 46 NAME=ORDERS OLDNAME=DISK2 SERIAL=659Ø46 BP=659Ø44
```

Then, to make the disk available for system use again, you ready the disk:

```
RY PK 46
```

You enter a PER PK command to verify the changes:

```
-------------------------------- PK STATUS -----------------------------
44*B        [659Ø44] #1 ORDERS (14)
46*C        [659Ø46:659Ø44:44] #2 ORDERS (2)
```

For a description of the PER PK display, see "Displaying Device Status (PER)" in this section.

Scanning Disks (SCAN)

The SCAN (Scan Disk or Pack Volume) system command is useful if you are having read/write problems with a disk, or if you think you have some bad sectors on a disk. Note that the disk is not available for system use during this procedure.

The SCAN command reads a disk volume, and analyzes and records any read errors. If there are errors, it generates DAMAGEDFILES and DAMAGEREPORT report files describing what sectors are bad, and what files are affected. For a full description of these files, see the REPLACE command entry in the *System Commands Reference Manual.*

If any error occurs in the read, you should notify your system administrator.

You can mark the bad sectors so that the system does not use them with the RES (Reserve) system command. See the System Commands Reference Manual for information about this command.

Procedure

The disk you want to scan must be reserved with a UR command before you enter the SCAN command. You also must close the disk (it cannot be in use during the scan).

Do the following to scan a disk:

1. Save the disk so that no new files can be opened on the disk. Enter

   ```
   SV PK <unit number>
   ```

   The unit number is the number of the disk you want to scan.

2. Wait until there are no files open on the disk.

   The SV command prevents new files from opening, but the files that are already open remain available until the programs using them close them.

   You can find out when there are no files open on the disk with the PER PK command. If you are not familiar with this command, see "Displaying Device Status (PER)" in this section.

3. Close the disk by entering

   ```
   CLOSE PK <unit number>
   ```

   If you are not familiar with the CLOSE system command, see "Closing Disk Units (CLOSE)" in this subsection.

4. Reserve the disk.

   The disk must be reserved before you can scan it. Enter

   ```
   UR PK <unit number>
   ```

5. Enter the SCAN command, with the following syntax:

   ```
   SCAN PK <unit number>
   ```

   The results of the disk scan are output in the system messages. You see a message at every 10 percent of the disk scan (10%, 20%, 30%, and so on) and after all of the disk is scanned, giving you the percentage of the disk that was read successfully.

   If not all of the disk is read successfully, talk with your system administrator. For information about the system administration issues involved in having bad disk sectors, see the *System Administration Guide*.

If you want to make the disk available for system use again, do the following:

1. Unreserve the disk by entering

   ```
   UR- PK <unit number>
   ```

2. Ready the disk by entering

   ```
   RY PK <unit number>
   ```

   If you are not familiar with the RY system command, see “Readying a Device (RY)” in this section.

The disk should again be available to the system. You can verify this with the OL (Display Labels and Paths) system command. Use of this command is described in “Displaying Labels and Paths (OL)” in this section.

Exception Event, Causing (HI)

An exception event is like an attention flag. Some programs are written so that you can interrupt the program to get or enter information. The HI (Cause EXCEPTIONEVENT) system command enables you to do this.

The HI command affects a process only if the program is written to accept HI commands, and the effect the commands have depend upon the particular program. If you buy software that accepts the HI command, it should have documentation telling you how and when to use it. If a programmer writes a program that accepts the HI command, he or she should let you know how and when to use it.

Some programs accept values with the HI command, and take action depending on the values. The program documentation or the programmer in charge of the program should tell you if this is the case, and what values to use.

An example of a program that accepts the HI command is LOGANALYZER. When LOGANALYZER is running, you can give it an exception event, and it tells you how much of the log it has analyzed, read, or sorted.

Procedures

- To use this command, find the mix number of the program you want to cause an exception event in, and enter

  ```
  <mix number> HI
  ```

  The response depends upon the program.

- To include a value with the HI command, enter

  ```
  <mix number> HI <value>
  ```

  The response depends upon the program.

File Access Structure, Rebuilding (RB)

The operating system uses the file access structure table (FAST) to locate the file addresses stored in a family's system directory. The FAST is ordered by filename so that the system can quickly locate files or subdirectories by name.

The FAST for each family is stored in the central system catalog or access directory.

If, for some reason, the FAST is out of date or becomes corrupted, the operating system might not be able to locate files on the disk family.

Procedure

To rebuild the FAST for a family, enter

```
RB ON <family name>
```

This command causes the operating system to read the entire system directory of a family and build a new FAST.

Files

This subsection contains basic information to help you handle both disk and tape files. For a definition of what a file is, and for file concepts, refer to “Basic File Concepts” in Section 1.

The following topics are covered in this subsection:

- Archiving disk files
- Backing up disk files
- Displaying file directories
- Displaying file information
- Copying disk and tape files
- Removing files from disk

Archiving Disk Files

A Series systems provide you with two methods of creating and tracking backup copies of disk files: the archiving subsystem and the cataloging subsystem. Only the archiving subsystem is discussed here. The cataloging subsystem is discussed in the *Disk Subsystem Guide.*

The archiving subsystem is designed to make it easy for you to make safety backup copies of disk files, to restore them when needed, and to move disk files to tape to free up disk pack space. The subsystem keeps track of the files that have been backed up or moved to tape. Note that to use the archiving subsystem, either you need a privileged usercode, or you must enter commands to the subsystem from the ODT.

The archiving subsystem tracks files through special directories it creates for each online disk family. These directories are stored on the DL CATALOG family. The archiving subsystem uses these directories to record the status and location of each file that has been backed up or moved to tape with an *ARCHIVE* WFL statement. Then, if you need to restore one of those files, the subsystem can tell you where the file is located.

The archiving subsystem has three categories of functions:

- Backup operations (moving files from disk to tape)
- Restore operations (moving files from tape to disk)
- Merge operations (moving files from tape to tape)

Backup operations are usually handled through WFL jobs, which are written by your system administrator or by a system programmer. You simply run the WFL job, and then supply appropriate tapes when asked by the system (these WFL jobs can even start automatically each evening). You are likely responsible, however, for restore and merge operations, and for freeing disk space by copying files to tape and removing them from the disk (this is called a "rollout" operation). The restore, merge, and rollout operations are described under "Procedures," later in this subsection.

To give you a basic understanding of the archiving subsystem, all the *ARCHIVE* WFL statements are explained in the following table. For more information on operations other than restore, merge, and rollout, see the *Disk Subsystem Guide.*

You access the archiving subsystem through the *ARCHIVE* WFL statement. Table 8–1 shows the ARCHIVE statements available to handle file archiving for your site.

Table 8–1. Archiving Subsystem ARCHIVE Options

ARCHIVE Option	Description
FULL	Copies all the resident disk files regardless of whether archive backup copies of those files already exist.
DIFFERENTIAL	Copies the resident disk files that either have changed since the last ARCHIVE FULL or ARCHIVE ROLLOUT request, or do not have an archive backup directory record.
INCREMENTAL	Copies the resident disk files that either have changed since the last ARCHIVE backup or rollout operation, or do not have an archive backup directory record.
ROLLOUT	Copies the selected files to tape and then removes them from the disk pack.
MERGE	Groups previously archived backup files from a given disk family onto a single tape set. This statement does not copy disk files, but merges files from different tapes onto a single tape set.
RESTORE	Copies the files from archive backup tapes to disk, even if there are versions of some or all the files already resident on the disk pack.
RESTOREADD	Copies the files from archive backup tapes to disk, unless there are versions of the files already on the disk pack. Only those files for which there are no conflicting versions already on disk are copied.
PURGE	Removes the archive backup records for the specified files or file directories from the archive directory. The PURGE statement does not affect any disk files, only archive directory records.

Each archive statement (except ARCHIVE PURGE) has three special options:

- & COMPARE
- & VERIFY
- & DSONERROR

The following information briefly explains each archive statement option:

- & COMPARE

  If you include this option in an archive statement, the archiving subsystem compares each file it copies with the source file after each file is copied. If any mismatches are detected during the compare process, library maintenance displays a message at the ODT asking whether the files created during the operation should be purged.

- & VERIFY

  If you include this option in an archive statement, the archiving subsystem verifies with a checksum each copy made during the operation. If any copy errors are detected during the verification, the subsystem displays a message at the ODT asking whether the files copied during the operation should be purged.

- & DSONERROR

  If you include this option in an archive statement, and an error occurs during processing, the archiving subsystem discontinues the operation it is performing. The subsystem then purges any tape it is creating and any information it stored in the archive directory during the operation. This option is used most frequently with the BACKUP or MERGE option. Use this option if you want to repeat the entire job sequence, and do not want a partial copy.

These options are more fully explained in the *WFL Reference Manual.*

Note that tapes created by the archiving subsystem are standard library maintenance tapes and are fully compatible with the COPY and ADD statements. Thus, if a backup tape is created by the archiving subsystem, you can copy files from the tape by using the ARCHIVE RESTORE, COPY, or ADD statement. In the discussion of the archive subsystem, all tapes referred to are standard library maintenance tapes.

Restoring Files with the Archiving Subsystem

The following ARCHIVE statements offer you different ways to restore previously archived files to a disk family:

- ARCHIVE RESTORE

  This statement restores all the requested files from tape, even if there are already versions of some or all of the files resident on disk.

- ARCHIVE RESTOREADD

  This statement restores the requested files from tape, unless there are already versions of the files resident on disk.

Use the ARCHIVE RESTORE statement to copy files from archive backup tapes to disk. You specify the disk family name and the names of the files and directories, or both, to be restored. The archiving subsystem looks up the information for the requested files in the archive directory for the disk family. The subsystem then makes a copy request and supplies the appropriate tape names and serial numbers for the requested files. You will then have to load each of the requested tapes.

Note that if your site has the AUTORESTORE option set, the archiving subsystem automatically restores backup files. When a program or WFL job has a NO FILE condition, there are archive records for the file, and the usercode of the process is the owner of the file. The subsystem requests that you load a specific tape, and supplies the tape name and serial number. When the tape is loaded and ready, the archiving subsystem proceeds to copy the file back to its previous usercode and pack family. Automatic restore operations can happen often if you have recently performed an ARCHIVE ROLLOUT operation.

Procedure

To restore files from an archive tape whether a version of the file is currently resident on disk, use the following syntax:

```
ARCHIVE RESTORE <statement option> <file title> TO <family name>
```

The statement option is one of the archive statement options (& DSONERROR, & VERIFY, or & COMPARE), and is optional. The file title is the title (usercode and file name) of the file you are restoring. The family name is the name of the disk family to which you are restoring the file.

Examples

Suppose a user has accidentally removed all the files under his usercode from the family DATABANK, and wants you to recover them for him. Since your site archives all changed files every night, you can do this for him. If the user's usercode is ROBERT, you enter the following archive statement:

```
ARCHIVE RESTORE (ROBERT)= TO DATABANK
```

This statement restores all the files under the usercode ROBERT to the disk family DATABANK. If some or all of his files are critical, you can ensure that all the files are restored error free by including the archive statement option & COMPARE in the statement as follows:

```
ARCHIVE RESTORE & COMPARE (ROBERT)= TO DATABANK
```

If there are no files archived for the usercode ROBERT on the pack DATABANK, the archiving subsystem issues a message that no files are selected, and the operation finishes. If there are files under that usercode, the archiving subsystem creates an RSVP message that gives you the tape name and serial number for the tape containing the files, and requests that you mount the tape. Once you have mounted the tape, you see a system message appear for each file as it is copied from the tape to the disk.

If the user has removed only some of the files under his usercode, you can use the ARCHIVE RESTOREADD statement to restore his files. This statement works the same as the ARCHIVE RESTORE statement, but the archiving subsystem checks to see if each file it restores is already present on the disk family. If any generation of the file is already present, the archiving subsystem does not restore that file.

Merging Archived Files with the Archiving Subsystem

The ongoing process of archiving files for your site can produce many different tapes containing archived files. The MERGE statement combines archive backup copies of files for a given disk family onto one tape set. This can save tapes, and make keeping track of archive tapes easier.

The archiving subsystem updates its records to show that the files are on the new tape set. The subsystem still has records of the old tapes in the archive directory too, because the subsystem remembers the four most recent backup tapes for each archived file.

The ARCHIVE MERGE process does not copy any files from the designated disk family, only from old archive tapes. This is true even if there are new files on the disk family that have not yet been archived, and if existing disk files have been updated, but the new versions have not been archived. If you want to include new and updated (but not yet archived) files in the ARCHIVE MERGE output tape, the files must first be archived with an ARCHIVE FULL, ARCHIVE INCREMENTAL, or ARCHIVE DIFFERENTIAL statement.

If errors occur during the merge process, contact your senior operator or system administrator.

Procedure

To initiate the merge process for a given disk family, use the following syntax:

```
ARCHIVE MERGE FROM <family name> TO <tape name>
```

The family name is the name of the disk family for which you are merging the archive backup files. The tape name is the name you want the new tape to have. For example, to merge the files on the family OPERATIONS to a tape named OPSTAPE, you enter

```
ARCHIVE MERGE FROM OPERATIONS TO OPSTAPE.
```

The archiving subsystem requests as input some or all of the tapes containing files from previous ARCHIVE operations for the specified disk family. Each tape is requested twice: once while the output tape directory is being created, and once when the files are being copied. Load each tape on a tape drive as it is requested by the subsystem. When the appropriate tape is loaded, the library maintenance automatically copies the needed files from the tape.

If any of the requested input tapes are unavailable, you can use the OF (Optional File) system command to reply to the “NO FILE” RSVP message for that tape during the first phase of the merge process. None of the files on the missing tapes are copied to the

MERGE output tape. For these files, the archive directory continues to reference the old copies on the missing tapes.

Using ROLLOUT with the Archiving Subsystem

The term *rollout* refers to the archiving subsystem process of copying a file to tape and then removing the disk file. The ARCHIVE ROLLOUT command invokes this process, which is useful for freeing up disk space (for example, in response to a SECTORS REQUIRED message). The archiving subsystem keeps track of the location of the files that have been rolled out, and can find them for you when the files are needed.

Note that you should contact your system administrator to determine your site's policies regarding rolling out files before you perform any rollout operations.

When a file is copied to the backup tape by the rollout process, a record is stored in the archive directory for the disk family indicating that there is a backup copy of that file on the new backup tape. When a file is selected for rollout, if there is already an archive directory record referencing an existing backup tape for the resident version of the file, then the file is not recopied to the rollout backup tape, but is simply removed from the disk.

You specify a usercode or a list of usercodes for the ARCHIVE ROLLOUT command, and the number of sectors to be freed. The archiving subsystem then selects files from the usercodes to roll out. The subsystem selects files to roll out from the specified usercodes depending on the FILEKIND, LASTACCESSDATE, and SAVEFACTOR file attributes, and on the size of the file. A large file that has not been accessed for a long time and that has a low SAVEFACTOR value is most likely to be selected by the rollout procedure. Conversely, a small file that has been accessed recently and has a high SAVEFACTOR value is not likely to be rolled out. Note that certain files (such as GUARDFILES) are not eligible for rollout selection.

The archiving subsystem selects files to roll out from the specified usercodes until the number of sectors specified in the command has been freed.

If you do not specify any usercodes from which files are to be rolled out, then only nonusercoded files are considered. (The file names of nonusercoded files appear as **<file name>* – for example, **SYSTEM/COBOL*.) If you specify a list of usercodes (or asterisk (*), for nonusercoded files), then files are selected from the family under the usercodes you specified. You can also specify ALL USERS, and then the rollout process considers all files on the disk family, including nonusercoded files.

If you use the words *ALL FILES* instead of a number of sectors for the rollout amount in the ARCHIVE ROLLOUT statement, all the files under the specified usercodes are copied and removed. This form of the command is useful if a usercode is no longer valid; you can make sure the files are saved and that they are removed from disk at the same time.

Procedures

Note that only a few basic forms of the ARCHIVE ROLLOUT statement are shown below. See the *Disk Subsystem Guide* for a more complete explanation of this statement.

- To roll out all the files for a particular usercode, use the following syntax:

  ```
  ARCHIVE ROLLOUT ALL FILES SELECT (<usercode>) FROM <family name>
       TO <tape name>
  ```

  This statement archives and removes all the files under the specified usercode on the given disk family.

- To free up a given amount of disk sectors from a disk family by archiving and removing files selected from all usercodes, use the following syntax:

  ```
  ARCHIVE ROLLOUT <number> SECTORS SELECT ALL USERS FROM
       <family name> TO <tape name>
  ```

  The number is the number of disk sectors to be cleared by the rollout operation. The family name is the disk family on which sectors are to be cleared. The tape name is the name of the output tape to be assigned to the operation.

- To clear a given number of disk sectors from a disk family by archiving and removing files from a specific usercode or list of usercodes, use the following syntax:

  ```
  ARCHIVE ROLLOUT <number> SECTORS SELECT (<usercode1>),(<usercode2>),...
       FROM <family name> TO <tape name>
  ```

  The number is the number of disk sectors to be cleared by the rollout operation. The usercode or usercodes are the usercodes to be considered by the rollout operation. Each usercode must be enclosed in parentheses. Multiple usercodes must be separated by commas. The family name is the disk family from which sectors are to be cleared, and the tape name is the name of the output tape for the operation.

Example 1

Suppose an employee has moved to a different department, and no longer uses his usercode (JOHN) on a system. He still has a large number of files under the usercode however, and these files might be needed later by someone else.

To make efficient use of the disk pack space on the system, you want to remove his files. Since the files still might be useful, you want to save them, and keep track of where they are so you can get them again if needed. So, you decide to use the archive subsystem to roll out all the files under his usercode. His disk family name is PRODUCTION.

You enter the following ARCHIVE ROLLOUT statement to accomplish this task:

```
ARCHIVE ROLLOUT ALL FILES SELECT (JOHN) FROM PRODUCTION TO
     JOHNSTAPE (SERIALNO = "JOHNS")
```

The archiving subsystem requests that you load the scratch tape that has the serial number JOHNS. Once you have loaded the tape, library maintenance begins the rollout procedure. As the rollout proceeds, the archiving subsystem displays a message for each file that is copied and removed, just as in regular library maintenance.

Example 2

Suppose you get a SECTORS REQUIRED ON OPS message at the ODT. To free up some disk sectors, you enter the following statement:

```
ARCHIVE ROLLOUT 10000 SECTORS SELECT ALL USERS FROM OPS TO ROLLOPS
```

The rollout operation selects 10,000 sectors worth of files from all usercodes and rolls them out to tape.

If one of the rolled out files is needed later, you can use the archiving subsystem to locate the file and restore it to disk.

You should check with your system administrator before performing any rollout procedures.

Backing Up Files

A backup file is an extra copy of a file, usually stored on tape. The term *backing up files* refers to the process of making copies of files (usually to tape) in case the originals are corrupted or lost.

It is extremely important that your site back up important disk files. What files are backed up, and how often, are decisions for the system administrator. As an operator, it is your responsibility to see that the correct files are backed up at the proper times.

To make the backup process easy and quick, Unisys provides the system utility FILECOPY. You can use the FILECOPY utility to make backup tape copies of disk files in addition to, or in place of, ARCHIVE backup procedures. The FILECOPY utility enables you to back up files based upon the following criteria:

- When files were last updated
- When files were last accessed
- When files were created

You can also choose files that do not meet the given criteria to be included in or excluded from the backup.

The contents of the FILECOPY task request determine which files are backed up. A task request is simply a set of instructions to the utility, defining the file selection criteria, the files location, and the destination to which they are to be copied.

A FILECOPY job can contain from 1 to 5 task requests. FILECOPY jobs have the following format:

```
RUN SYSTEM/FILECOPY;
EBCDIC CARD
<task request>;
      .
      .
      .
<task request>
?END JOB
```

The procedures in this subsection describe the basic task requests for backing up files meeting the criteria in the preceding list. The FILECOPY utility has many useful features not described here, such as the ability to include and exclude files. You can also use defaults for things such as the tape name and serial number. For extensive information on the FILECOPY utility, see the *A Series System Software Utilities Manual*.

Files Accessed during a Specific Time Period

You can back up only the files that have been accessed during a specific period (before or after a certain time, or between two times). To do this, use the FILECOPY ACCESSED task request in a FILECOPY run as described in the following procedure.

The FILECOPY task request described in this procedure backs up all the files accessed in the time period you enter, on the disk you specify. You can also name the tape the files are copied to, or use the default (in this case the default tape name is ACCESSED).

To back up the accessed files on more than one disk, you can include multiple FROM specifications. You can also combine more than one FILECOPY task request in a single FILECOPY run by separating the task requests with a semicolon (;).

To modify the task request (include files not accessed in the specified time, for example) or create a more advanced task request, refer to the *A Series System Software Utilities Manual* for information.

Procedure

This procedure copies all the files to tape that have been accessed in the time period you specify. The FILECOPY task request in this procedure backs up files for one disk. This task request also creates an index file containing the file names of all the files that were copied:

```
RUN SYSTEM/FILECOPY;
EBCDIC CARD
ACCESSED <before/after/between specification>
FILES (= FROM <disk name> TO <tape name>(KIND=TAPE, SERIALNO=
("<serial number>")) LOCKINDEX
INDEXLABEL <file name>
?END JOB
```

The *before/after/between specification* is where you specify the time period for choosing accessed files. You have the following choices:

- BEFORE mm/dd/yy @ nnnn
- AFTER mm/dd/yy @ nnnn
- BETWEEN mm/dd/yy @ nnnn - mm/dd/yy @ nnnn

In the preceding list, *mm/dd/yy* represents the date (04/15/92, for example), and *nnnn* represents the time using the 24-hour clock, without the colon (2330 is 11:30 p.m., for example). Zero represents midnight. You can leave out the date and time, and the default values are today's date, 12:00 a.m. Entering AFTER alone, for example, means all files accessed after 11:59 (2359) last night.

The disk name is the name of the disk you are backing up files from. If you want to back up files from multiple disks, separate the FROM specifications with commas:

```
(= FROM <disk name>, = FROM <disk name>, = FROM <disk name>)
```

The tape name is the name of the tape you are copying the backup files to, and the serial number is the serial number of that tape. Use the SN (Serial Number) system command to purge the tape you are backing up files to, and to assign the tape a name and serial number. This command is described in “Tapes” in this section.

The LOCKINDEX command tells FILECOPY to create an index file containing the file names that were copied. The INDEXLABEL command lets you name the index file, and the file name is the name you choose for it. The index file is written to the disk where the files being backed up reside.

Examples

The following example backs up all the files accessed after 7:00 a.m. on May 30, 1989, on the disk named PRODUCTION. The backup files are copied to a tape named ACCESSED, and the tape’s serial number is 53089.

```
RUN SYSTEM/FILECOPY;
EBCDIC CARD
ACCESSED AFTER 5/3Ø/92 @ Ø7ØØ
FILES (= FROM PRODUCTION) TO ACCESSED (KIND=TAPE, SERIALNO =
("53Ø92"))
?END JOB
```

The next example backs up the files accessed between 7:00 a.m. and 5:30 p.m. on 5/30/92, on the disks named PAYROLL, DISK, and INVENTORY. The files are copied to a tape named BACKUP, and the tape’s serial number is 1234. This task request also creates an index file named ACCESSED/INDEX.

```
RUN SYSTEM/FILECOPY;
EBCDIC CARD
ACCESSED BETWEEN 5/3Ø/92 @ Ø7ØØ - 5/3Ø/92 @ 173Ø
FILES (= FROM PAYROLL, = FROM DISK, = FROM INVENTORY)
TO BACKUP(KIND=TAPE, SERIALNO =("1234")) LOCKINDEX
INDEXLABEL ACCESSED/INDEX
?END JOB
```

The final example combines the previous two examples in the same FILECOPY run. This is an example of combining task requests together.

```
RUN SYSTEM/FILECOPY;
EBCDIC CARD
ACCESSED AFTER 5/3Ø/92 @ Ø7ØØ
FILES (= FROM PRODUCTION) TO ACCESSED (KIND=TAPE, SERIALNO =
("53Ø92"));
ACCESSED BETWEEN 5/3Ø/92 @ Ø7ØØ - 5/3Ø/92 @ 173Ø
FILES (= FROM PAYROLL, = FROM DISK, = FROM INVENTORY)
TO BACKUP(KIND=TAPE, SERIALNO =("1234")) LOCKINDEX
INDEXLABEL ACCESSED/INDEX
?END JOB
```

Files Created during a Specific Time Period

You can back up the files that have been created during a specific period (before or after a certain time, or between two times). To do this, use the FILECOPY CREATED task request in a FILECOPY run as described in the following procedure.

The FILECOPY task request described in the procedure backs up all the files created in the time period you enter, from the disk you specify. You can also name the tape the files are copied to, or use the default (in this case, the default name is CREATED).

To back up the created files from more than one disk, you can include multiple FROM specifications. You can also combine more than one FILECOPY task request in a single FILECOPY run by separating the task requests with a semicolon (;).

To modify the task request (include files not created in the specified time, for example) or create a more advanced task request, refer to the *A Series System Software Utilities Manual* for information.

Procedure

This procedure copies the files to tape that have been created in the time period you specify. The FILECOPY task request in this procedure backs up files for one disk. This task request also creates an index file containing the file names of the files that were copied:

```
RUN SYSTEM/FILECOPY;
EBCDIC CARD
CREATED <before/after/between specification>
FILES (= FROM <disk name> TO <tape name>(KIND=TAPE, SERIALNO=
("<serial number>")) LOCKINDEX
INDEXLABEL <file name>
?END JOB
```

The *before/after/between specification* is where you specify the time period for choosing created files. You have the following choices:

- BEFORE mm/dd/yy @ nnnn
- AFTER mm/dd/yy @ nnnn
- BETWEEN mm/dd/yy @ nnnn - mm/dd/yy @ nnnn

In the preceding list, *mm/dd/yy* represents the date (04/15/92, for example), and *nnnn* represents the time using the 24-hour clock, without the colon (2330 is 11:30 p.m., for example). Zero represents midnight. You can leave out the date and time, and the default values are today's date, 12:00 a.m. Entering AFTER alone, for example, means all files created after 11:59 (2359) last night.

The disk name is the name of the disk you are backing up files from. If you want to back up files from multiple disks, separate the FROM specifications with commas:

```
(= FROM <disk name>, = FROM <disk name>, = FROM <disk name>)
```

The tape name is the name of the tape you are copying the backup files to, and the serial number is the serial number of that tape. Use the SN (Serial Number) system command to purge the tape you are backing up files to, and to assign the tape a name and serial number. This command is described in "Tapes" in this section.

The LOCKINDEX command tells FILECOPY to create an index file containing the file names that were copied. The INDEXLABEL command lets you name the index file, and the file name is the name you choose for it. The index file is written to the disk where the files being backed up reside.

Examples

The following example backs up all the files created after 7:00 a.m. on May 30, 1992, on the disk named PRODUCTION. The backup files are copied to a tape named CREATED, and the tape's serial number is 53089.

```
RUN SYSTEM/FILECOPY;
EBCDIC CARD
CREATED AFTER 5/3Ø/92 @ Ø7ØØ
FILES (= FROM PRODUCTION) TO CREATED (KIND=TAPE, SERIALNO =
("53Ø92"))
?END JOB
```

The next example backs up the files created between 7:00 a.m. and 5:30 p.m. on 5/30/92, on the disks named PAYROLL, DISK, and INVENTORY. The files are copied to a tape named BACKUP, and the tape's serial number is 1234. This task request also creates an index file named CREATED/INDEX.

```
RUN SYSTEM/FILECOPY;
EBCDIC CARD
CREATED BETWEEN 5/3Ø/92 @ Ø7ØØ - 5/3Ø/92 @ 173Ø
FILES (= FROM PAYROLL, = FROM DISK, = FROM INVENTORY)
TO BACKUP(KIND=TAPE, SERIALNO =("1234")) LOCKINDEX
INDEXLABEL CREATED/INDEX
?END JOB
```

The final example combines the previous two examples in the same FILECOPY run. This is an example of combining task requests together.

```
RUN SYSTEM/FILECOPY;
EBCDIC CARD
CREATED AFTER 5/3Ø/92 @ Ø7ØØ
FILES (= FROM PRODUCTION) TO CREATED (KIND=TAPE, SERIALNO =
("53Ø92"));
CREATED BETWEEN 5/3Ø/92 @ Ø7ØØ - 5/3Ø/92 @ 173Ø
FILES (=FROM PAYROLL, = FROM DISK, = FROM INVENTORY)
TO BACKUP(KIND=TAPE, SERIALNO =("1234")) LOCKINDEX
INDEXLABEL CREATED/INDEX
?END JOB
```

Files Updated during a Specific Time Period

You can back up the files that have been updated during a specific period (before or after a certain time, or between two times). To do this, use the FILECOPY UPDATED task request in a FILECOPY run as described in the following procedure.

The FILECOPY task request described in the procedure backs up all the files updated in the time period you enter, from the disk you specify. You can also name the tape the files are copied to, or use the default (in this case, the default name is UPDATED).

To back up the updated files from more than one disk, you can include multiple FROM specifications. You can also combine more than one FILECOPY task request in a single FILECOPY run by separating the task requests with a semicolon (;).

To modify the task request (include files not updated during the specified time period, for example) or create a more advanced task request, refer to the *A Series System Software Utilities Manual* for information.

Procedure

This procedure copies the files to tape that have been updated in the time period you specify. The FILECOPY task request in this procedure backs up files for one disk. This task request also creates an index file containing the file names of the files that were copied:

```
RUN SYSTEM/FILECOPY;
EBCDIC CARD
UPDATED <before/after/between specification>
FILES (= FROM <disk name> TO <tape name>(KIND=TAPE, SERIALNO=
("<serial number>")) LOCKINDEX
INDEXLABEL <file name>
?END JOB
```

The *before/after/between specification* is where you specify the time period for choosing updated files. You have the following choices:

- BEFORE mm/dd/yy @ nnnn
- AFTER mm/dd/yy @ nnnn
- BETWEEN mm/dd/yy @ nnnn - mm/dd/yy @ nnnn

In the preceding list, *mm/dd/yy* represents the date (04/15/92, for example), and *nnnn* represents the time using the 24-hour clock, without the colon (2330 is 11:30 p.m., for example). Zero represents midnight. You can leave out the date and time, and the default values are today's date, 12:00 a.m. Entering AFTER alone, for example, means all files updated after 11:59 (2359) last night.

The disk name is the name of the disk you are backing up files from. If you want to back up files from multiple disks, separate the FROM specifications with commas:

```
(= FROM <disk name>, = FROM <disk name>, = FROM <disk name>)
```

The tape name is the name of the tape you are copying the backup files to, and the serial number is the serial number of that tape. Use the SN (Serial Number) system command to purge the tape you are backing up files to, and to assign the tape a name and serial number. This command is described in "Tapes" in this section.

The LOCKINDEX command tells FILECOPY to create an index file containing the file names that were copied. The INDEXLABEL command lets you name the index file, and the file name is the name you choose for it. The index file is written to the disk where the files being backed up reside.

Examples

The following example backs up all the files updated after 7:00 a.m. on May 30, 1992, on the disk named PRODUCTION. The backup files are copied to a tape named UPDATED, and the tape's serial number is 53092.

```
RUN SYSTEM/FILECOPY;
EBCDIC CARD
UPDATED AFTER 5/30/92 @ 0700
FILES (= FROM PRODUCTION) TO UPDATED (KIND=TAPE, SERIALNO =
("53092"))
?END JOB
```

The next example backs up the files updated between 7:00 a.m. and 5:30 p.m. on 5/30/92, on the disks named PAYROLL, DISK, and INVENTORY. The files are copied to a tape named BACKUP, and the tape's serial number is 1234. This task request also creates an index file named UPDATED/INDEX.

```
RUN SYSTEM/FILECOPY;
EBCDIC CARD
UPDATED BETWEEN 5/30/92 @ 0700 - 5/30/92 @ 1730
FILES (= FROM PAYROLL, = FROM DISK, = FROM INVENTORY)
TO BACKUP(KIND=TAPE, SERIALNO =("1234")) LOCKINDEX
INDEXLABEL UPDATED/INDEX
?END JOB
```

The final example combines the previous two examples in the same FILECOPY run. This is an example of combining task requests together.

```
RUN SYSTEM/FILECOPY;
EBCDIC CARD
UPDATED AFTER 5/30/92 @ 0700
FILES (= FROM PRODUCTION) TO UPDATED (KIND=TAPE, SERIALNO =
("53092"));
UPDATED BETWEEN 5/30/92 @ 0700 - 5/30/92 @ 1730
FILES (= FROM PAYROLL, = FROM DISK, = FROM INVENTORY)
TO BACKUP(KIND=TAPE, SERIALNO =("1234")) LOCKINDEX
INDEXLABEL UPDATED/INDEX
?END JOB
```

Displaying File Directories

You can list the names of files on your disks either at the ODT with the PD (Print Directory) system command or at a printer with the DIR (Directory) system command.

If you are not familiar with A Series file and directory naming conventions, see "Disk File Naming Conventions" in Section 1 of this book.

Displaying Directories at the ODT (PD)

You can list the files on your disks at the ODT with the PD (Print Directory) system command.

Procedure

You can display any directory level with the PD command. The basic syntax is

```
PD <directory>/= ON <family name>
```

This command displays all the files in a directory. That is, you can display all the files with the same first node with this command. The equals sign (=) means *any node.*

So if you want to check files under the directory SYSTEM,use *PD SYSTEM/= ON PACK* to display all the files with the first node SYSTEM on the PACK family.

If you want to look at all the files on a family with the same first two (or more) nodes, enter *PD <directory>/<directory>/= ON <family name>*.

You can display all the files under a specific usercode with the following PD syntax:

```
PD (<usercode>)= ON <family name>
```

If you enter a complete file name in the PD command, the system responds with the following information about the file: date and time of creation, last access, last alteration, file size, and file security.

Examples

Assume you have these files on the family INVENTORY:

```
JANUARY/FORMS/INVENTORY
JANUARY/FORMS/INVENTORY/ONORDER
JANUARY/OVERLAND/SHIPPING/RECORDS
JANUARY/PERISHABLE/INVENTORY
MISCELLANEOUS/RECORDS
A/B/C
A/B/C/D
B/C/D
B/E/F
D
```

If you want to display all the directories on the family (list the first nodes of the files), enter the following command:

```
PD = ON INVENTORY
```

The system responds with the following list:

```
A (DIRECTORY)
B (DIRECTORY)
D (ALGOLCODE)
JANUARY (DIRECTORY)
MISCELLANEOUS (DIRECTORY)
```

Note that if several files have the same directory (the same first node), the directory is displayed only once.

If you want to list all the files under the directory JANUARY, you enter

```
PD JANUARY/= ON INVENTORY
```

The system displays the following list:

```
JANUARY (DIRECTORY)
. FORMS (DIRECTORY)
. . INVENTORY (DATA)
. . . ONORDER(DATA)
. OVERLAND (DIRECTORY)
. . SHIPPING (DIRECTORY)
. . . RECORDS (DATA)
. PERISHABLE (DIRECTORY)
. . INVENTORY (DATA)
```

To look at the files under the directory JANUARY/FORMS, enter

```
PD JANUARY/FORMS/= ON INVENTORY
```

The system displays the following list:

```
JANUARY (DIRECTORY)
. FORMS (DIRECTORY)
. . INVENTORY (DATA)
. . . ONORDER (DATA)
```

Printing Directories on the Printer (DIR)

If you want a hard copy report of the files on your disks, use the DIR (Directory) system command. This command does not generate screen display.

The DIR command generates a printout that lists the files under a given directory or for a given usercode. The following examples show you how to generate printouts listing files for a given usercode or directory, or listing all files for a disk family.

The DIR command invokes the utility SYSTEM/FILEDATA, which can generate a variety of reports. For more information on this utility, see the *A Series System Software Utilities Manual.*

Procedures

- To print a list of everything under a directory, enter

```
DIR FILENAMES: DIRECTORY = <directory name> FAMILY = <family name>
```

- To print all the nonusercoded files on a family, enter

```
DIR <family name>
```

Examples

If you want a list of all the files in the directory (USER1)PROGRAMS on the family USERDISK, you enter

```
DIR FILENAMES: DIRECTORY = (USER1)PROGRAMS FAMILY = USERDISK
```

If you want a list of all the files and their disk sector assignments on the family INVENTORY, you enter

```
DIR INVENTORY
```

Displaying File Information

You can use the PD (Print Directory) system command to display the following information about a file:

- File kind
- Creation date
- Last access date
- Last change date
- File size
- File security status
- Backup information (archive and catalog)

The security status of a file can be one of the following:

Security	Access
Public	The file can be accessed by all users.
Private	The file is accessible to the usercode under which the file was created, and to privileged users.
Guarded	The file is under protection of a guard file. A guard file is a file created by the GUARDFILE utility program, and describes the access rights of various users and programs to a program or file. The GUARDFILE utility is an optional security feature available with Data Management System II (DMSII) software.
Controlled	A controlled file has different kinds of access rights designated by the SECURITYUSE file attribute. Controlled file status is available with DMSII software.
LOCKEDFILE	Disk files with the LOCKEDFILE file attribute set to TRUE are protected from being accidentally removed, recopied, or having their names changed. Tape files with the LOCKEDFILE file attribute set to TRUE cannot be accidentally purged.

Default file security is private. The usage can be read only, write only, read/write, or execute only. The default usage value for files is read/write.

How file security status affects you depends upon the privileges of the usercode you use at the ODT (talk to your supervisor to find out). If the usercode is *privileged*, then you can access any file on the system. If the usercode is *nonprivileged*, then you can list and access public files and only the guarded and controlled files your usercode has been given access to.

For more information on file security, and system security in general, ask your system administrator.

Procedure

To display file information, use the following syntax:

```
PD <file name> ON <family name>
```

Example

To look at information for the file JANUARY/FORMS/INVENTORY, you enter

```
PD JANUARY/FORMS/INVENTORY ON INVENTORY
```

The system displays the following information:

```
FILE JANUARY/FORMS/INVENTORY ON INVENTORY (DATA)
DATE AND TIME OF CREATION: Sunday June 09, 1992 (92160)  AT 23:37:22
              LAST ACCESS: Friday June 14, 1992 (92165)  AT 10:24:59
              LAST  ALTER: Sunday June 09, 1992 (92160)  AT 23:37:56
TOTAL SECTORS: 52920 (504 PER AREA, CRUNCHED)
SECURITY: PUBLIC - USAGE: READ ONLY       LOCKEDFILE

ARCHIVE ENTRY 0 (DATA)
   CYCLE : 1   VERSION: 0
     TIMESTAMP:  FRIDAY JUNE 14, 1992 AT 18:30:38
     LASTACCESS: FRIDAY JUNE 14, 1992 AT  9:40:38

ARCHIVE FULL TO:  UTAPE
   TAPE SERIALNO = (061492)
   BACKUP STARTED ON: FRIDAY JUNE 14, 1992 (92165) AT 19:30:07
```

Copying Files (COPY, ADD)

Use the Work Flow Language (WFL) COPY and ADD statements to initiate a process named *LIBRARY/MAINTENANCE to copy files between disks and tapes. You can enter the statements at the ODT just as you do system commands. The following is a list of the COPY and ADD statements you can use and a description of what each statement does.

ADD	Copies files to disk and does not overwrite existing files
ADD & COMPARE	Same as ADD but compares copied files
ADD & VERIFY	Same as ADD but uses checksums to test correctness of copy
COPY	Copies disk files to and from disk or tape
COPY & COMPARE	Copies disk files to and from disk or tape and compares them
COPY & VERIFY	Copies disk files to and from disk or tape and uses checksums to test correctness of copy

The COPY statement copies files between disks and tapes. It does not change the original file in any way.

The ADD statement is similar to the COPY statement. The ADD statement does not overwrite; it does not copy a file if a file with the same name is already on the disk you are copying to. The ADD statement skips the file and continues. This statement is useful if you want to add a directory to a disk when some of the files are already there.

The COMPARE option compares the copied file to the original file to make sure no errors occurred during the copy. If there has been a mismatch, you get a "RECOPY REQUIRED" RSVP message.

Comparing files takes up resources and time, and is necessary only if the file is critical (such as a backup file directory). You should use COMPARE when copying to tape, however.

The VERIFY option causes library maintenance to calculate checksums for each file copied. When a file is copied to disk the procedure rereads the file from that disk and compares the checksum. If there is a mismatch a "RECOPY REQUIRED" RSVP message is issued.

When you copy files to tape with the VERIFY option, the library writes a checksum on the tape after each file copied. When all the files are copied to a tape, the library maintenance rewinds that tape, reads each file from the tape while calculating a new checksum for each file. Library maintenance compares the read checksums with those written on the tape. If there are any mismatches, library maintenance stops and asks you if the tape is to be purged, the file ignored, or the process terminated by the DS (Discontinue) command.

If you want to copy all the files from a disk or tape, use an equal sign (=) in place of the file name.

Only the basic COPY and ADD statements are explained here. For a detailed explanation of all the possible options, see the *WFL Reference Manual*.

Note: *The following examples copy unprotected files from disk to disk and disk to tape. If you want to overwrite a disk or tape file protected by the LOCKEDFILE file attribute, you must enter the WFL* ALTER *command to reset the LOCKEDFILE file attribute before you can remove or overwrite the file. For more information about setting and resetting the LOCKEDFILE file attribute, see "Possible Copying Problems" later in this section, as well as the* File Attributes Reference Manual. *For more information about the WFL* ALTER *command, see the* WFL Reference Manual.

Copying Files from Disk to Disk

Use the following instructions to copy a file from disk to disk (and overwrite any file with the same name on the destination disk). In the following syntax, the command can be ADD, ADD & COMPARE, COPY, or COPY & COMPARE.

```
<command> <file name> FROM <disk name>(KIND=DISK) TO
<disk name>(KIND=DISK)
```

Example

The following example copies the file FILE1 from the disk named DISK1 to the disk named DISK2:

```
COPY FILE1 FROM DISK1(KIND=DISK) TO DISK2(KIND=DISK)
```

If you want the new copy to have a different filename than the source file, use the following syntax:

```
<command> <file name> AS <file name> FROM <disk name>(KIND=DISK) TO
<disk name>(KIND=DISK)
```

Example

The following example copies FILE1 to DISK2 with the new filename X. This command works exactly the same as the preceding COPY command, but it makes the name of the copied file the name you specify after AS.

```
COPY FILE1 AS X FROM DISK1(KIND=DISK) TO DISK2(KIND=DISK)
```

This command does not change the name of the original file.

Copying Files to or from Tapes

When you copy to or from a tape, you need to mount and load the appropriate tape on a tape drive, and make the drive online. Refer to the tape drive user manual for instructions.

The first file on a library maintenance tape is a directory of all the disk files copied to that tape. Because of this, when you request a copy to a tape, library maintenance copies files to the tape starting at the beginning of the tape. You cannot add files to the end of a tape. Each time you copy to a tape, library maintenance erases what is already there, and starts the copy at the beginning of the tape with a new tape directory of the files you are copying.

You can copy many files to the same tape, but you must do this with one copy statement.

There are many outside variables that can interfere with your copying task. For example, if another program or user on the system has entered a copy command before you, the system can take your tape for that other copy as soon as you make the tape available.

You can prevent other jobs from using your tape with an option called SERIALNUMBER (Option 27). When this option is set, every job or program that writes to tape must specify the serial number of the tape volume to be used. Not only does setting this option prevent other jobs from using your tapes, but it also ensures that you are copying to the tape you intend to be the destination tape.

To set this option, enter the following at the ODT:

```
OP+ 27
```

The system responds by telling you the option is set. For more information about this and other MCP options, refer to "Setting and Resetting MCP Options (OP)" later in this section.

Note that when you use tapes, you should rewind them with the RW (Rewind) system command (instead of rewinding them from the tape drive). This enables the operating system to check if a job is using the tape, and prevents aborted reads and writes to and from tapes.

You can instruct the operating system to automatically unload tapes from a unit after a reel switch or file close releases the unit by setting the AUTOUNLOAD option of the MODE (Unit Mode) system command.

To set this option, enter the following command:

```
MODE MT <unit name> AUTOUNLOAD ON
```

If you set AUTOUNLOAD to OFF, the operating system unloads tapes automatically only if you specify a file close that locks the unit on which the tape is mounted. To set AUTOUNLOAD to OFF, enter the following command:

```
MODE MT <unit name> AUTOUNLOAD OFF
```

You can use the AUTOUNLOAD option in programs and in the WFL statements COPY, ADD, and ARCHIVE to specify whether the operating system should unload a tape. The default value for AUTOUNLOAD is OFF. See the *System Commands Reference Manual* for more information on the MODE system command.

Copying Files from Disk to Tape

Use the following syntax to copy a file from disk to tape:

```
COPY <file name> FROM <disk name>(KIND=DISK) TO
<tape name>(SERIALNO = "<serial number>")
```

Example

The following example copies the file named FILE1 from the disk named DISK1 to a tape with the serial number MYTAPE.

```
COPY FILE1 FROM DISK1(KIND=DISK) TO
TAPE1(SERIALNO = "MYTAPE")
```

Following are step-by-step instructions for copying a file from a disk to a reel-type tape. For this procedure, the ODT must be in automatic display mode (as described in "Monitoring Processes at the ODT" in Section 3).

1. Get the tape you want to copy to. Remember, you will lose the present contents of the tape.
2. Put a write ring on the tape.
3. Find an available tape drive.

   To look at what is on the tape drives right now, enter:

   ```
   PER MT
   ```

   If any tapes are mounted and ready, you see a display similar to the following:

   ```
   -----MT STATUS----
   28*P   [ØØØ116] 16ØØ #1 1:Ø <12/Ø7/9Ø> MASTER/TEST
   29 P   S C R A T C H
   ```

   This shows you the tape labels for the tapes loaded and ready on each drive.

   To look at all the tape drives available to the system, including those presently offline, enter:

   ```
   PER MT-
   ```

   If there are tapes on the drives, you see something like this:

   ```
   ------------------------------- MT STATUS ------------------------------
   28*P   [ØØØ116] 16ØØ #1 1:Ø <12/Ø7/9Ø> MASTER/TEST
   29 P   S C R A T C H
   3Ø P   625Ø     NOT READY
   ```

   For an explanation of these displays, refer to "Displaying Device Status (PER)" earlier in this section.

4. Load the tape on an available tape drive. Refer to the user manual for your tape drives for instructions.

5. Use the SN (Serial Number) system command to purge the tape and assign a serial number to it, or the PG (Purge) system command to erase the tape contents without changing the tape name and serial number. See *Note* below.

   Note: *Some computer departments assign each new tape a permanent serial number. This serial number remains with the tape throughout its usage. If your computer department uses tapes with permanent serial numbers, do not use the SN (Serial Number) system command to purge a tape and change the tape serial number. Instead, use the PG (Purge) system command. The PG system command erases the tape contents, but does not change the tape name or erase the tape serial number. For more information on the Purge command, see "Purging Tapes (PG)" later in this section.*

   To use the SN command, enter

   ```
   SN MT <unit number> <serial number>
   ```

   The unit number is the number of the tape drive you are using. The serial number can be any six digits, letters, or combination of both.

   For example, if you want to purge the tape on drive 28 and assign the tape the serial number MYTAPE, enter

   ```
   SN MT28 MYTAPE
   ```

   For an explanation of the SN (Serial Number) command, refer to "Assigning Serial Numbers to Tapes (SN)" in this section.

   After you enter the command, the system responds with the following message:

   ```
   MT28 WILL BE SN-ED
   ```

   This means that the operation will be attempted.

   If the SN is successful, the system displays the following message:

   ```
   MT28 PURGED
   ```

   If the SN is not successful, the tape files might have been protected with the LOCKEDFILE file attribute. Look for an operator RSVP message similar to the following in the waiting entries, asking you to confirm the tape purge:

   ```
   MT28 CONFIRM PURGE MASTER/TEST [ØØØ116]
   ```

   You have the option of accepting the purge command, or cancelling the purge. You decide to accept, so you enter:

   ```
   <mix number> OK
   ```

   The tape is purged and ready to use for copying files.

6. Enter *PER MT* again. The listing for the unit number of the tape drive you are using should show the serial number you specified and a status of SCRATCH. The display looks like this:

```
--------------------------- MT STATUS ----------------------------------
 MT28*P    [MYTAPE]  16ØØ   S C R A T C H
```

If it does not, see "Possible Copying Problems" following the copying tasks.

7. Enter the COPY command, using the following copy syntax:

```
COPY <file name> FROM <disk name>(KIND=DISK) TO
<tape name>(SERIALNO="<serial number>")
```

You can name the tape anything you like. The tape name you specify in the copy command is the name of the tape until new files are copied to the tape, the tape is purged, or another program writes a file on the tape.

8. Find out when your job finishes.

If the file you are copying is a large one, the copy might take a few minutes. While the copy is in progress, you can see your job in the active entries. Enter the A (Active Entries) system command to see the active entries. Look for your job number.

When your job finishes, it shows up in the system messages and the completed entries. Track the system messages as they appear under the *Messages* entries on the ODT screen.

Look for your job number. When your job finishes, you see a message similar to the following:

```
------Mix----Time----------Message--------------------------------------
     2274    10:13       FILE1 COPIED FROM DISK1 TO TAPE1
```

Your copy is now complete.

If you do not get this message (if you get a "NO FILES COPIED" message, for example) then your copy was not successful. The *Messages* entries display indicates what is wrong. If you do not understand what happened, see the "Possible Copying Problems" subsection after the copying tasks.

9. Rewind the tape by entering the RW (Rewind) system command as follows:

```
RW MT <unit number>
```

The unit number is the number of the tape drive you are using.

10. Remove your tape from the tape drive, and remove the write ring from the tape.

Copying Files from Tape to Disk

To copy a file from tape to disk, use the following syntax:

```
<command> <file name> FROM <tape name>(SERIALNO =
"<serial number>") TO <disk name>(KIND=DISK)
```

Example

The following example copies the file FILE1 from the tape TAPE1 to the disk DISK1:

```
COPY FILE1 FROM TAPE1(KIND=TAPE, SERIALNO = "MYTAPE")
TO DISK1(KIND=DISK)
```

Following are step-by-step instructions for copying a file from a tape to a disk. For this procedure, the ODT must be in automatic display mode (as described in "Monitoring Processes at the ODT" in Section 3).

1. Get the tape containing the files you want to copy.
2. Find an available tape drive.

   To look at what is on the tape drives right now (if anything), enter

   ```
   PER MT
   ```

   If there is anything on the drives, you see a display similar to the following:

   ```
   -----MT STATUS----
   28*P   [ØØØ116] 16ØØ #1 1:Ø <12/Ø7/9Ø> MASTER/TEST
   29 P   S C R A T C H
   ```

   This shows you what the tape header is for the tapes loaded on each drive.

   To look at the tape drives presently offline, but available to the system, enter

   ```
   PER MT-
   ```

   If there are any, you see something like this:

   ```
   ------------------------------- MT STATUS -----------------------------
   28*P   [ØØØ116] 16ØØ #1 1:Ø <12/Ø7/9Ø> MASTER/TEST
   29 P   S C R A T C H
   3Ø P   625Ø    NOT READY
   ```

   For an explanation of these displays, refer to "Displaying Device Status (PER)" in this section.
3. Mount and load the tape on an available tape drive. Refer to the user manual for your tape drives for instructions.
4. Check the tape to see if it is the one you want. Enter

   ```
   PER MT
   ```

   You should see a display similar to the following:

   ```
   ------------------------------- MT STATUS -----------------------------
        28 P  [MYTAPE]  (2293)  16ØØ  #1  1:Ø  <8/Ø1/9Ø> TAPE1/FILEØØØ
   ```

   The tape drive number is in the first field. The tape serial number is enclosed in square brackets. The job number is in the field to the right of the serial number. The tape name is in the last field, to the left of the slash (in this example, the tape name is TAPE1).
5. Enter the COPY command, using the following syntax:

   ```
   <command> <file name> FROM <tape name> TO
   <disk name>(KIND=DISK)
   ```
6. Find out when your job finishes.

When your job finishes, a message appears in the message entries on the ODT screen. Track the system messages as they appear in the *Messages* entries on the ODT screen.

If the file you are copying is a large one, the copy might take a few minutes. While the copy is in progress, you can see your job in the active entries listing on the ODT screen.

When your job does finish, a message entry similar to the following appears:

```
---Mix-Time--------------------- MESSAGES -----------------------------
  7698 14:34    FILE1 COPIED FROM TAPE1 TO DISK1
```

Your copy is now complete.

If you do not get this message (if you get a "NO FILES COPIED" message, for example) then your copy was not successful. You can find out what the problem was through the system messages. If you cannot find out what was wrong, see "Possible Copying Problems" following the copy tasks.

7. Rewind the tape by entering the RW (Rewind) system command as follows:

```
RW MT <unit number>
```

   The unit number is the number of the tape drive you are using.

8. Remove your tape from the tape drive.

Copying Files from Tape to Tape

To copy a file from a library maintenance tape to another tape, use the following syntax:

```
COPY <file name> FROM <tape name> TO
<tape name>(SERIALNO="<serial number>")
```

Example

The example copies the file FILE1 from the tape named TAPE1 to the tape named TAPE2.

```
COPY FILE1 FROM TAPE1 TO TAPE2(SERIALNO="MYTAPE")
```

Following are step-by-step instructions for copying a file from one tape to another. For this procedure, the ODT must be in automatic display mode (as described in "Monitoring Processes at the ODT" in Section 3).

1. Get the tape you want to copy from (the source tape), and the tape you want to copy to (the destination tape). Remember, you will lose what is presently on your destination tape during the copy procedure.
2. Put a write ring on the destination tape (and make sure there is no write ring on the source tape).
3. Find two available tape drives. If you have only one, copy the files on the source tape to a disk, and then copy the files from the disk to the destination tape.

To look at what is on the tape drives right now (if anything), enter

```
PER MT
```

If there is anything on the drives, you see a display similar to the following:

```
-----MT STATUS----
28 P    [ØØØ116] 16ØØ #1 1:Ø <12/Ø7/9Ø> MASTER/TEST
29*P    S C R A T C H
```

This shows you what the tape header is for the tapes loaded on each drive.

To look at the tape drives presently offline, but available to the system, enter

```
PER MT-
```

If there are any, you see something like this:

```
------------------------------- MT STATUS ------------------------------
28 P   [ØØØ116] 16ØØ #1 1:Ø <12/Ø7/9Ø> MASTER/TEST
29*P   S C R A T C H
3Ø P   625Ø    NOT READY
```

For an explanation of these displays, refer to "Displaying Device Status (PER)" in this section.

4. Mount and load the tapes on available tape drives. Refer to the user manual for your tape drives for instructions.

5. Use the SN (Serial Number) system command to purge the tape and assign a serial number to it.

   Note: *If your computer department assigns permanent serial numbers to its tapes, use the PG (Purge) system command to purge the contents of the tape without changing the tape serial number. For more information on how to purge a tape, see "Purging Tapes (PG)" later in this section.*

   To use this command, enter

   ```
   SN MT <unit number> <serial number>
   ```

   The unit number is the number of the tape drive where you mounted your destination tape. The serial number can be any six digits, letters, or combination of both.

   For example, if you want to purge the tape on drive 28 and assign the tape the serial number MYTAPE, enter

   ```
   SN MT28 MYTAPE
   ```

   For an explanation of the SN command, see "Assigning Serial Numbers to Tapes (SN)" in this section.

   After you enter the command, the system responds with the following message:

   ```
   MT <unit number> WILL BE SN-ED
   ```

   This message means that the operation will be attempted.

If the SN is successful, the system displays the following message:

```
MT<unit number> PURGED
```

If the SN is not successful, the tape files might have been protected with the LOCKEDFILE file attribute. Look for an operator RSVP message similar to the following in the waiting entries, asking you to confirm the tape purge:

```
MT<unit number> CONFIRM PURGE <tape name> [serialno]
```

You have the option of accepting the purge command, or cancelling the purge. You decide to accept, and enter:

```
<mix number> OK
```

6. Enter the PER MT command again. The listing for the unit number of the tape drive you are using should show the serial number you specified and a status of SCRATCH. The display looks like this:

```
--------------------------- MT STATUS -----------------------------------
    MT28*P    [MYTAPE]  1600   S C R A T C H
```

   If it does not, see “Possible Copying Problems” following this list.

7. Verify your source tape.

   The PER MT command you entered for the last step also shows you what is on your source tape. Look for the entry by the unit number of the tape drive you mounted your source tape on. Make sure this is the tape you want.

8. Enter the COPY command, using the following syntax:

```
COPY <file name> FROM <tape name> TO
<tape name>(SERIALNO="<serial number>")
```

   You can get the source tape name from the PER MT display. It is in the last field on the line, to the left of the slash. You can name your destination tape anything you like; the tape name you specify for the destination tape becomes the name of that tape volume.

9. Find out when your job finishes.

   When your job finishes, a message appears in the message entries on the ODT screen. Track these messages to see when your job finishes.

   When it does, you see a message entry similar to the following:

```
---Mix-Time--------------------- MESSAGES ------------------------------
  7698 14:34     FILE1 COPIED FROM TAPE1 TO TAPE2
```

   Your copy is now complete.

   If you do not get this message (if you get a “NO FILES COPIED” message, for example) then your copy was not successful. You can find out what the problem was in the system messages. If you cannot figure out what was wrong, see “Possible Copying Problems” following the copy tasks.

10. Rewind the tapes by entering the RW(Rewind) system command as follows:

```
RW MT <unit number>
```

The unit number is the number of the tape drive you are using.

11. Remove your tapes from the tape drives, and remove the write ring from the destination tape.

Possible Copying Problems

This topic deals with errors commonly introduced during the copy procedure.

Following are example system responses to common errors. Each system message is followed by an explanation of the cause of the message and a procedure for correcting the problem.

Message

```
---Mix-Time--------------------- MESSAGES -----------------------------
  8201 10:13     MT28 IS NOT WRITE ENABLED
```

Cause

There is no write ring on the tape loaded on MT28, and a copy to the tape was requested. You must put a write ring on any tape you want to write to or purge. You do not need a write ring for tapes you want to copy from.

Procedure for Correcting the Problem

1. Rewind the tape, using the RW (Rewind) system command as described in the copy procedures.
2. Remove the tape from the tape drive.
3. Put a write ring on the tape.
4. Mount and load the tape on the drive again. Make the drive online.
5. Start your copy procedure again.

Message

```
---Job-Task--Pri--Elapsed-------- 1 WAITING ENTRY ---------------------
  8206/8207   50   1:14     REQUIRES PK XNAME
```

Cause

You might have tried to copy to or from a disk that does not exist or is offline. You also see this message if you misspelled a disk name.

If you spelled the disk name correctly, but the disk is offline, the job waits until the disk is online again, then finishes executing normally. If you do not want to wait, you can discontinue the job as described in the following procedure.

Procedure for Correcting the Problem

1. Discontinue the job using by entering the DS (Discontinue) system command as follows:

   ```
   <job number> DS
   ```

The job number is listed under *Job* in the waiting entries list on the ODT screen. To discontinue the job in the preceding message example, enter:

```
8206 DS
```

Use the job number, not the task number, for the DS command.

After you do this, the waiting entry message should go away for that job number. If it does not, check the job number, and enter the DS command again. Look in the Messages entries for the system response.

2. Reenter your copy command, making sure to spell the disk name correctly.

Message

```
---Job-Task-Pri---Elapsed------- 1 WAITING ENTRY -----------------------
  7009\7069  50      1:19 (SUE) *LIBRARY/MAINTENANCE
       DUP FILE(LOCKEDFILE):(SUE)STATUSREPORT ON PACK PK247
```

Cause

You tried to overwrite a disk file protected by the LOCKEDFILE file attribute. Protected disk and tape files cannot be overwritten, removed, or changed without first resetting the LOCKEDFILE file attribute to FALSE. To reset this attribute, you need to have a privileged usercode.

Procedure for Correcting the Problem

To reset the LOCKEDFILE file attribute for the protected file to FALSE, enter the following at the ODT:

```
WFL ALTER (SUE)STATUSREPORT(LOCKEDFILE=FALSE)
```

The operating system responds with a message similar to the following:

```
#7088 WFL: ONE FILE ALTERED
```

After you have reset the LOCKEDFILE attribute to FALSE, you can overwrite or remove protected disk and tape files.

Message

```
NO UNIT CORRESPONDENCE
```

Cause

You entered a nonvalid unit number in a command. For example, if you have two tape drives, MT28 and MT29, and you enter the command SN MT92, you get this message.

Procedure for Correcting the Problem

Reenter the command with a valid unit number.

Message

```
8211 CONTROL CARD ERROR
  8197  5Ø  :Ø2  JOB*SYSTEM/PRINT/ROUTER
```

Cause

This message, which is displayed near the top of the ODT screen, is caused by a syntax error in your COPY command. A common reason for this message is missing quotation marks around the serial number specification. If you have a serial number that contains any letters, you must enclose it in quotation marks in the COPY command. Refer to the example syntax.

Procedure for Correcting the Problem

Reenter the COPY command, paying close attention to the spelling, punctuation, and order of text in the statement. If your tape serial number begins with a letter, remember to enclose the serial number in quotes in the COPY command.

Message

```
---Job-Task--Pri--Elapsed---------- 1 WAITING ENTRY --------------------
  8214/8215   5Ø    :47       *LIBRARY/MAINTENANCE
       TAPE2/FILEØØØ REQUIRES MT[TEST] #1
```

Cause

You specified a serial number incorrectly, or the tape unit is not ready, or the tape is not write-enabled. The preceding message indicates that a COPY command was entered, requesting a copy to TAPE2, and the serial number TEST was specified.

Procedure for Correcting the Problem

1. Make sure the correct tape volume is online and write-enabled. Use PER MT to check the status of the tapes. Make sure that an asterisk (*) appears in the display, indicating the tape is write-enabled. Discontinue the copy job by entering the DS (Discontinue) system command as follows:

   ```
   <job number>DS
   ```

   To discontinue the job in the preceding example message, you would enter

   ```
   8214 DS
   ```

Use the job number for the DS command, not the task number.

After you do this, the waiting entry message should go away for that job number. If it does not, check the job number, and enter the DS command again. Look in the Messages entries for the system response.

2. Reenter your COPY command with the correct serial number.

Messages

```
---Mix-Time--------------------- MESSAGES ----------------------------
  8220 10:32     MT30 PURGED
  8219 10:32     NO FILES COPIED
  8219 10:32     TESSFILE NOT ON TESTDISK
```

Cause

The file TESSFILE on the disk TESTDISK is not available. You see similar messages if you mistype a file name you are copying from a disk to a tape, or if the file you are trying to copy is not on the disk you specify in the copy command.

Procedure for Correcting the Problem

Reenter the COPY command with the correct file name. If the file name is correct and you still get these messages, look on another disk for the file you want. See “Displaying Directories at the ODT (PD)” under “Files” in this section if you do not know how to do this.

Message

```
---Job-Task--Pri--Elapsed-------------- 1 WAITING ENTRY ----------------
  8222/8223   50    1:50      *LIBRARY/MAINTENANCE
       NO FILE PACKX/FILE000(MT) #1
```

Cause

You mistyped the name of the tape from which you are copying, the tape loaded on the drive is not the tape you think it is, or you wanted to copy from a disk named PACKX but you forgot to put (KIND=PACK) after the disk name. Library maintenance assumes that you are trying to copy your file to a tape unless you specify (KIND=DISK) in your copy statement.

If you enter a command to copy from tape before you mount and load the tape, a waiting entry like the preceding one appears.

Procedure for Correcting the Problem

Check your copy command for typing errors in the tape name. If you find one, do the following:

1. Use PER MT to determine if the tape is online and to verify its name. Discontinue the job waiting for the misspelled tape name. Enter the DS (Discontinue) system command:

   ```
   <job number> DS
   ```

   The job number is the first number on the line for your waiting entry (8222 in the preceding example).

   After you do this, the waiting entry message should go away for that job number. If it does not, check the job number, and enter the DS command again. Look in the Messages entries for the system response.

2. Reenter the copy command with the correct tape name.

If you have mounted the correct tape, but you still have a waiting entry, the name of the tape on the label might not match the name of the tape on the tape drive.

- Check the system messages to see if the job is requesting a particular volume of a tape set. You might have mounted volume #2 of TAPW2 when the job is asking for volume #1.
- Use PER MT to check the volume number of the tape on the drive. The volume number appears after the serial number and tape size, preceded by a number sign (#).
- Use PER MT to verify that the name of the tape on the drive and the tape name in the waiting job entry are the same.

If the information on the tape is not what is listed on the tape label, do the following:

- Unload the tape from the drive.
- Correct the tape label to reflect the tape contents.
- Mount and load the next tape in the series, and make the drive online.

If this is the correct tape, the system completes the copy (you see messages to that effect). You can look at the tape header to see what the tape name is by entering

```
PER MT
```

If it is not the one you want, you can repeat this procedure, or just discontinue the job, using step 1 in the preceding list.

If you have entered the copy command before you loaded the appropriate tape, simply load that tape, make the drive online, and the copy will commence.

Removing Files (REM)

You can remove files from disks with the Work Flow Language (WFL) REMOVE statement.

You can remove one file at a time, a list of files, a directory, or a list of directories (or any combination). You can specify any family to remove from.

If the file you want to remove has been protected with the LOCKEDFILE file attribute, you cannot remove it without first using the WFL *ALTER* command to reset the LOCKEDFILE attribute. The procedure for resetting the LOCKEDFILE file attribute is described under "Procedures" later in this subsection. Refer to the *File Attributes Reference Manual* for more information about the LOCKEDFILE file attribute. For more information about the WFL *ALTER* command, see the *WFL Reference Manual.*

Note: *Once you remove a file, it is gone, and you cannot get it back unless you have a backup copy of the file. If you remove a file that is currently in use, the file is marked as removed when you enter the command, but is not physically removed until the file is no longer being used. In this case, the file is not visible after the REMOVE statement is entered, even though it is physically present until it is no longer in use.*

For a more in-depth explanation of the REMOVE statement, refer to the *WFL Reference Manual.*

Procedures

- To remove a single file, enter

```
REMOVE <file name> FROM <family name>
```

- To remove a list of files, enter

```
REMOVE <file name>,<file name> FROM <family name>
```

  You can list any number of files in this command.

- To remove a directory, enter

```
REMOVE <directory name>/= FROM <family name>
```

  Note that if the directory name also represents a file, that file is not removed by this command. For example, assume you have the following files:

```
TEST
TEST/ONE
TEST/TWO
```

The name TEST represents both the directory name and a file. If you entered the command *REMOVE TEST/=*, the file TEST would be left. If the directory name does not also represent a file, all files are removed. For example, if you had the following files, and you entered the same command, no file would remain:

```
TEST/ONE
TEST/TWO
TEST/THREE
```

- To remove a group of directories, enter

```
REMOVE <directory name>/=,<directory name>/= FROM <family name>
```

  You can list any number of directories in this command.

- To remove a disk file that has been protected with the LOCKEDFILE file attribute, first reset the LOCKEDFILE attribute to FALSE. To do this, you need to have a privileged usercode. At the ODT, or through CANDE or MARC, enter

```
WFL ALTER (SUE)STATUSREPORT(LOCKEDFILE=FALSE)
```

  The system responds with

```
#76Ø5 WFL: ONE FILE ALTERED
```

  You can now remove the file.

Examples

To remove the file X from the family USERDISK, enter

```
REMOVE X FROM USERDISK
```

To remove the files A/B, C, and D from the family USERDISK, you would enter

```
REMOVE A/B,C,D FROM USERDISK
```

To remove all the files under the directory PROGRAM from the family DISK, enter

```
REMOVE PROGRAM/= FROM DISK
```

To remove all the files in the directories PROGRAM, DATA, and STUFF from the family TESTDISK, enter

```
REMOVE PROGRAM/=, DATA/=, STUFF/= FROM TESTDISK
```

Halt/Load, Performing (??PHL)

A halt/load temporarily interrupts system operation and loads the master control program (MCP) from the halt/load family into main memory.

If the MCP in main memory becomes corrupted for some reason, and the system begins working incorrectly (for example, the ODT quits responding), a halt/load might fix the problem.

If a program goes into an infinite loop and you cannot discontinue it, a halt/load might fix the problem.

A halt/load takes about 4 minutes, and the system should return to an idle running state. If it does not, you might have a more serious problem. See "Initializing the System" for information on where to find detailed initialization procedures for your system.

The system has full halt/load recovery capabilities. That is, the system can return to an idle state on its own if everything is working correctly. The ability of individual programs to recover, however, depends upon how the programs were written.

WFL jobs restart from the last point where no tasks were running. Files that were open under CANDE recover fully when reopened.

Other jobs or tasks might lose data or fail to recover properly. For example, suppose a payroll program is not designed to keep track of what it has processed as it runs. If the system halt/loads, when the job is restarted, each person whose paycheck had been processed before the program was interrupted will be paid twice.

A halt/load procedure should be considered a drastic measure and should be used only as a last resort. If at all possible, give users warning and enough time to stop their programs before the halt/load. Talk to your system administrator about pre- and post-halt/load procedures.

The system uses a file called the job description file for system state recovery (the file is named "*JOBDESC"). This file contains information on all the files and programs on the system. As long as this file is not removed or corrupted, a halt/load causes minimal information loss.

Procedure

To perform a halt/load, enter

```
??PHL
```

The halt/load occurs immediately.

Halt/Load Units, Creating Alternate (CM)

The halt/load unit on your system is the disk or family containing the currently operative operating system code file. It has been designated to the system as the halt/load unit either during system initialization or with the HLUNIT (Halt/Load Unit) system command.

You should create a standby halt/load unit as a backup in case the primary halt/load unit fails. If this happens, you can use the alternate unit to restart and run the system. Refer to the *System Administration Guide* for details and considerations involving alternate halt/load units.

A standby halt/load unit is a disk or family containing a valid copy of the operating system that has been identified to the system with the CM (Change MCP) system command.

When you create a standby halt/load unit, the MCP's states and tables and the configuration file are saved on the family. This happens when you enter the CM system command in the following procedure. Then, if you use the standby unit to halt/load, the system reinitializes exactly as it was when you were last using the original halt/load family.

You can find out if a disk or family is an alternate halt/load unit by entering the following command:

```
CM? ON <family name>
```

If the disk or family is an alternate unit, the system responds with the name of, and information about, the MCP code file on that unit. If the disk or family is not an alternate unit, the system responds with the message "THAT FAMILY IS NOT CM'ED."

Procedure

Use the following procedure to create an alternate halt/load unit:

1. Copy the MCP code file to the disk or family you intend to use as the alternate halt/load family.

   If you do not know how to copy a file from disk to disk, see "Copying Files (COPY, ADD)" in this section.

2. Use the following form of the CM (Change MCP) system command to make the disk or family capable of halt/loading:

   ```
   CM <file name> ON <family name>
   ```

   The file name is the name of the MCP code file you copied, and the family name is the name of the disk or family you want as the alternate halt/load unit.

3. Use the following form of the CM system command to make the standby disk or disk family capable of halt/loading:

```
CM STANDBY ON <family name>
```

The disk or family is now an active standby halt/load unit.

Jobs and Tasks

The procedures in this subsection deal with jobs and tasks: a *job* is one or more tasks under the control of a single Work Flow Language (WFL) program. Each job has a mix number and looks similar to tasks in system displays. A *task* is a single, complete unit of work performed by the system, such as compiling or executing a program, or copying a file from one disk to another. Tasks are initiated by jobs. Note that the term *process* refers to both jobs and tasks.

For system operations, you can usually treat jobs and tasks in the same way. They both have their own mix numbers, and they look similar in ODT displays.

In the following topics, if there is need for a distinction, the procedure points it out.

Changing Job and Task Priority (PR)

Priority is a job or task characteristic that determines its precedence in the use of system resources. The higher the priority, the more likely it is that the job or task will execute quickly.

The priority of a job or task can range from 1 through 99, where 99 is the highest priority. If no priority is specified, the default priority is 50.

When you give a job or task a high priority, it makes that job or task run faster, but it also makes all the jobs and tasks with a lower priority run that much slower. For a discussion of the system management considerations involved in job and task priority, see the *System Administration Guide.*

You can use the PR (Priority) system command to change priority values of jobs and tasks. Changes in priority from a PR command are temporary; they do not last across halt/loads.

Procedure

To change job or task priority, use the following syntax:

```
<mix number list> PR <number>
```

The mix number list specifies the mix numbers of one or more jobs or tasks you want to change the priority of. If you are changing more than one, separate the mix numbers with a comma. The number is the priority number, and must be between 1 and 99, inclusive.

Example

To change the priorities of the jobs with mix numbers 4972 and 4980 to 75, enter

```
4972,4980 PR 75
```

Displaying Job and Task Status (Y)

The Y (Status Interrogate) system command display can show you valuable information for a job or task. The following table describes each display element in the Y display:

Display element	Description
Originating job number	The job or task number is displayed in the form <job number>/<task number>, so you can find the originating job for the task in question.
Job priority	Job priority ranges from 1 to 99. If the priority is too low, a job or task runs slowly. If it is too high, a job or task runs quickly but causes other jobs to run slowly. The default value is 50.
Job origin	This information can help you find out who started a job if there is no usercode associated with it.
Usercode	This is the usercode of whoever started the job. There is no usercode if, for example, a program started the job.
Chargecode	This tells you what account the job is being charged to.
Program name	This is the name of the program. If the program is a library, the Y command returns a list of the programs using the specified mix number.
DISPLAY message associated with the job	This is the most recent message generated by the job for display.
RSVP message associated with the job	If there is an RSVP message in the Y display, the job needs some sort of input from you or a programmer. Your choices for a valid response are listed in the REPLY entry in the display.
Possible responses to an RSVP message	If the job has an RSVP message, the valid ways you can respond are listed on the REPLY line.

These are some of the common elements in the Y display, but there can be others. See the *System Commands Reference Manual* for more information about the Y command and its displays.

Procedure

To use the Y system command, enter

```
<mix number> Y
```

The mix number is the mix number of the job you want a status display for. You can find the mix number in the Job-Task column in the active entries display.

Example

Suppose a user calls you, wondering why her job is not running. She tells you that the name of her program is OBJECT/PRG1TEST.

You see the following waiting entry:

```
---Job-Task-Sub-Pri---Elapsed--- 1 WAITING ENTRY ----------------------
  4440/4443   3  50      6:31 (SARA) OBJECT/PRG1TEST ON TESTDISK
       OPERATOR STOPPED
```

The display says OPERATOR STOPPED. This means that someone has entered an ST (Stop) system command for the job. You want to know what commands you can use to get the program running again.

To find out, you enter

```
4443 Y
```

The resulting display looks similar to this:

```
STATUS OF TASK 4440/4443 AT 13:59:12
PRIORITY = 50
ORIGINATION: LSN 285
USERCODE: SARA
CHARGECODE: SOFTDEV
STACK STATE: WAITING ON AN EVENT
PROGRAM NAME: (SARA)OBJECT/PRG1TEST ON TESTDISK
RSVP: OPERATOR STOPPED
REPLY: OK,DS
```

The RSVP line tells you that the job has been stopped. The REPLY line tells you the valid commands you can enter to remove this job from the waiting entries.

Since you want to resume the job, you enter

```
4443 OK
```

Discontinuing Jobs and Tasks (DS)

Use the DS (Discontinue) system command to terminate jobs and tasks immediately. If a job or task is in an infinite loop, or if a user accidentally starts a program, this command terminates it.

The DS command is also helpful for removing unwanted jobs and tasks from queues, or for terminating scheduled or waiting jobs and tasks.

You cannot discontinue the operating system or other system-critical programs. You cannot discontinue locked programs. Locked programs must be unlocked with the LP– (Unlock Program) system command before you can discontinue them. The LP system command is described in detail in the *System Commands Reference Manual.*

Procedure

To terminate a job or task, enter

```
<mix number list> DS
```

The mix number list specifies one or more jobs or tasks you want to discontinue. You can find the mix numbers in the Job-Task column in the active entries. Separate mix numbers with commas. The system responds with an "OPERATOR DSED" message in the system messages. When you see the message, that job or task has been terminated.

Displaying Job and Task Time Information (TI)

The TI (Times) system command displays the current values of the time accumulators for jobs and tasks.

This information can tell you a number of things. It can tell you that a job or task is taking more time to process than normal. It also tells you how much processor or I/O time a job or task is getting. This type of information can help you, for example, to decide whether the job or task is executing, or if it is caught in a loop.

The following are the elements in a TI command display:

Display Element	Description
PROCESS	This is the amount of time the MCP has spent processing the job or task (such as executing instructions or performing calculations).
IO	This is the amount of I/O time the job or task has used.
READYQ	This is the amount of time the job or task has spent waiting for processor time. If this time is high in relation to the PROCESS time, then your system is handling a heavy load of processes with higher priorities, and the job is spending a lot of time waiting to be processed.
INITPBIT	This is the amount of time the operating system spent allocating memory areas and a count of the number of memory areas and code areas the system read back from the overlay file for the job or task.
OTHERPBIT	This is the amount of time the operating system has spent moving data areas for the job or task in and out of main memory. If this number is high, then the system's memory resources are scarce, and the operating system is spending a lot of time managing memory instead of processing jobs and tasks. If the OTHERPBIT time is high in relation to the other numbers in this display, you might want to look at the OTHERPBIT value for the entire system. You can do this with the U (Utilization) system command described in "Displaying System Utilization (U)" in this section. If that number is also high, your system's memory resources are under too much strain, and you should bring this to the attention of your system administrator.
ELAPSED	This is the total elapsed time since the job or task was initiated.

Procedure

To display the current values of the time accumulators for a job or task, enter

```
<mix number list> TI
```

The information is returned in hours, minutes, and seconds.

Example

If you want to see the TI display for a job with the mix number 5285, you enter

```
5285 TI
```

The system response looks like this:

```
TIMES FOR 5285

    PROCESS   = ØØ:ØØ:37
    IO        = ØØ:ØØ:Ø1
    READYQ    = ØØ:ØØ:56
    INITPBIT  = ØØ:ØØ:Ø6        365 operations
    OTHERPBIT = ØØ:ØØ:Ø2        68  operations
    ELAPSED   = ØØ:11:4Ø
```

Forcing Job and Task Initiation (FS)

The FS (Force Schedule) system command starts scheduled or queued jobs or tasks if there is enough available memory.

If there is enough memory, jobs or tasks are unconditionally started, whether they are in a queue or scheduled by the operating system.

If there is not enough memory, you see the message

```
INSUFFICIENT MEMORY TO FULFILL FS REQUEST
```

In this case, the job or task remains scheduled.

The FS command is valuable when you want a job or task in a queue to start as soon as possible. This command is also useful when an HS (Hold Schedule) system command is in effect, and you need to initiate a specific job or task. The HS command is described in “Preventing Job and Task Initiation (HS)” in this section.

Procedure

To force a job or task to start, enter

```
<mix number list> FS
```

The mix number list specifies the one or more jobs or tasks you want to start.

Example

If you want to force the job with mix number 7852 to start, you enter

```
7852 FS
```

Preventing Job and Task Initiation (HS)

Use the HS (Hold Schedule) system command to prevent new jobs and tasks from entering the mix or to enable new jobs and tasks to enter the mix again.

An example of the use of this command is in the "Switching DATACOMINFO Files" task in this section. Use this command when it is important that no new jobs or tasks enter the mix.

You can use the FS (Force Schedule) system command to start specific jobs and tasks when the HS command is in effect. The FS command is described in "Forcing Job and Task Initiation (FS)" in this subsection.

Procedures

- To stop new jobs and tasks from entering the mix, enter

  ```
  HS
  ```

  The system responds with the message

  ```
  JOB SELECTION STOPPED
  ```

- To allow new jobs and tasks into the mix again, enter

  ```
  HS-
  ```

  The system responds with the message

  ```
  JOB SELECTION RESUMED
  ```

- To display the current status of the HS command, enter

  ```
  HS?
  ```

  If HS is enabled (no new jobs can enter the mix), the response is "SCHEDULE IS HELD." If HS is disabled, the response is "SCHEDULE IS NOT HELD."

Passing Text to Jobs and Tasks (AX)

The AX (Accept) system command enables you to send text to a program (job or task), usually in response to that program requesting input with an "ACCEPT" message.

Some programs are designed to accept input from the ODT, and then take action on that input. When a program requires this sort of input, a request for input appears in the waiting entries with an ACCEPT message. When you pass your response to the program with the AX command, the program leaves the waiting entries and resumes processing.

Not all programs are written to accept input this way, and generally you need to know what the appropriate replies for a program's ACCEPT message are. The person responsible for the program should tell you what needs to be passed to the program.

Procedure

To pass text to a program waiting with an ACCEPT message, use the following syntax:

```
<mix number> AX <user text>
```

The mix number is the mix number of the job or task you want to pass the user text to. The user text is the information to be passed to the job or task.

You can find the mix number of the program by looking in the Job-Task column in the waiting entries display.

Example

Suppose you run a program at the end of every week that generates an inventory report. The program is written so that the report can go to the printer, to tape, or to both. When the program has finished creating the report, it comes up in the waiting entries with an "ACCEPT" message, asking you which type of output you want:

```
---Job-Task--Pri--Elapsed------------- 1 WAITING ENTRY ---------------
  1528/1528   5Ø    1:Ø3  JOB (SITE)INVENTORY/GENERATOR
              ACCEPT:OUTPUT TO PRINTER, TAPE OR BOTH
```

Your manager told you that valid replies to this ACCEPT message are PRINTER, TAPE or BOTH. You want the output to go to the printer, so you enter

```
1528 AX PRINTER
```

The job then leaves the waiting entries and sends a report to the printer.

Restarting Active Jobs and Tasks (RESTART)

The RESTART (Restart Active Jobs) system command restarts currently active jobs.

The RESTART command provides you with a way to manually discontinue an active job and restart it later from the point the job was ended, rather than from the beginning. Note that a halt/load has the same effect, but with the RESTART command, you do not have to interrupt the entire system, just the job or jobs you specify.

When you enter the RESTART command for a job, the system discontinues the job and initiates the job restart process. Before the job actually begins execution again, the operating system puts out a waiting entry for the job for which you must enter either an OK (Reactivate) command or a DS (Discontinue) command. You can leave the job in the waiting entries until you are ready for it to begin again.

The RESTART command is useful, for example, if you must take a disk pack offline for maintenance, but the disk pack has one or more large, long running jobs active on it. You can use the RESTART command to end the jobs and restart them after the pack is back online.

Note that the system restarts the job from the point of the last set of completed tasks. This means that dependent tasks the job started and that have completed will not be executed again. Dependent tasks running at the time the RESTART command was entered, however, are restarted from the beginning of the task.

Procedure

To restart an active job or list of active jobs, use the following syntax:

```
<mix number list> RESTART
```

The mix number list specifies one or more active jobs you want to restart. Separate mix numbers with a comma.

Example

Suppose a disk pack called INVENTORY has a faulty power supply, and needs to be taken offline for repair. However, a job that processes weekly transactions is currently running on that pack, and you would lose a lot of time if you had to start the job over. The job is called WEEK/TRANSACTIONS. You decide to stop the job with the RESTART command, so you can continue the job after the power supply is fixed. You enter

```
1234 RESTART
```

The number 1234 is the mix number of WEEK/TRANSACTIONS.

Resuming Suspended Jobs and Tasks (OK)

Use the OK (Reactivate) system command to resume jobs or tasks that have been stopped with a ST (Stop) system command or have been suspended for some reason.

If a job or a task has been stopped or suspended, the system displays it in the waiting entries, along with the reason it is waiting. To find out if the OK command can resume the job or task, use the Y (Status Interrogate) system command, described in "Displaying Job and Task Status (Y)" earlier in this subsection.

Procedure

To use the OK system command, enter

```
<mix number list> OK
```

The mix number list specifies one or more jobs or tasks you want to resume. Separate mix numbers with a comma.

Example

You see the following task in the waiting entries:

```
---Job-Task-Sub-Pri---Elapsed--- 1 WAITING ENTRY -----------------------
  4440/4443   3  50      6:31 (FRED) OBJECT/PRG1TEST ON TESTDISK
       OPERATOR STOPPED
```

The message associated with the task is "OPERATOR STOPPED." If the message says something about "stopped" or "suspended," the OK command is usually a valid choice.

To resume the task in this example, enter

```
4443 OK
```

Note that you enter the task number in this case, not the job number, because you want to resume the task.

Running WFL Jobs (START)

To run a Work Flow Language (WFL) job, you must use the START statement. Note that you cannot use the RUN statement to execute jobs written in WFL. The RUN statement is intended to execute compiled programs. The RUN statement is described in "Programs, Running (RUN)" in this section.

The START statement can be used from the ODT with or without a usercode associated with it. If the statement is used without a usercode, all the files associated with the job that was started must be nonusercoded or must have PUBLIC security status.

The following procedures describe how to execute a job and how to schedule a job to be executed at a certain time. For more information about the START statement, and about job initiation, see the *WFL Reference Manual*.

Procedures

- To initiate a WFL job, use the following syntax:

  ```
  START <file name> ON <family name>
  ```

  The file name is the name of the WFL job you want to initiate. The family name is the family that file is stored on. For example, to initiate the WFL job DAILY/BACKUP, which is stored on the family OPERATIONS, enter

  ```
  START DAILY/BACKUP ON OPERATIONS
  ```

  After the job starts, it is listed in the active entries. Any messages generated by the job are listed in the system messages.

- To schedule a WFL job to start at a specific time, use the following syntax:

  ```
  START <file name> ON <family name>; STARTTIME = <time>
  ```

  The file name is the file the WFL job you want to initiate is saved in. The family name is the family that the file is stored on. The time is the time you want the job to start (in 24-hour format). The time has the form *hh:mm ON mm/dd/yy*. The date is optional.

  For example, if you want to schedule the job DAILY/BACKUP to start today at 7:00 p.m., you enter

  ```
  START DAILY/BACKUP ON OPERATIONS;STARTTIME = 19:00
  ```

  If you want to schedule the same job to start at 7:00 p.m. on March 30, 1992, you enter

  ```
  START DAILY/BACKUP ON OPERATIONS;STARTTIME = 19:00 ON 3/30/92
  ```

Suspending Jobs and Tasks (ST)

Use the ST (Stop) system command to suspend a job or task from processing. This command is useful when you need to take a job or task out of the active entries, but do not want to discontinue the job or task.

A job or task suspended with an ST command is placed in the waiting entries with the message "OPERATOR STOPPED." The job or task can be restarted with the OK (Reactivate) system command. The OK command is described in "Resuming Suspended Jobs and Tasks (OK)" earlier in this subsection.

Procedure

To suspend a job or task, use the following syntax:

```
<mix number> ST
```

For example, if you want to suspend a job with the mix number 8457, you enter:

```
8457 ST
```

Example

Suppose there are several large jobs running on your system that require large amounts of processor time, and the combination of these jobs running together is slowing system response time. You do not want to discontinue any of the jobs, but you want to take one of them out of the mix until one of the other jobs finishes. You know that one of the jobs, INVENTORY/UPDATE, uses large amounts of processor time, so you decide to suspend that job. You look in the active entries for the mix number:

```
---Mix-Sub-Pri--CPU Time-------- 17 ACTIVE ENTRIES ---------------------
  117Ø   5  5Ø       :13 (USER1)OBJECT/TEST ON USERDISK
  1197   5  5Ø      2:28 STATISTICS
  7422   5  5Ø      1:18 INVENTORY/UPDATE ON INVENTORY
                             .
                             .
                             .
```

Using the mix number from the active entries display, you suspend INVENTORY/UPDATE with the ST command:

```
7422 ST
```

Job 7422 is suspended and placed into the waiting entries:

```
---Job-Task-Sub-Pri---Elapsed--- 1 WAITING ENTRY ----------------------
  7422\7422   5  5Ø       :34 INVENTORY/STATISTICS ON INVENTORY
     OPERATOR STOPPED
```

The job remains in the waiting entries until an OK command is entered for it. When this occurs, the job resumes processing.

Job Queues

Job queues are waiting lists from which the operating system selects jobs to enter the mix. You can display the queues on your system with the SQ (Show Queue) system command, described in "Displaying Job Queues and Queue Attributes (SQ, QF)" in this subsection.

Every job queue has several attributes. When selecting jobs to enter the mix, the operating system considers them in order of priority. Queue attributes control the way jobs are assigned to queues and the way jobs are selected from queues for execution.

You can make a queue using all the default values, or tailor the queue with queue attributes. Queue attributes enable you to establish default values for and place limits on the jobs using the queue, such as the number of jobs that can execute from that queue at any given time, the amount of processor time a job can use, the amount of memory, and so on. You can display the attributes of a queue with the QF (Queue Factors) system command, also described in "Displaying Job Queues and Queue Attributes (SQ, QF)" in this subsection.

Your system can have one default job queue, which you designate with the DQ (Default Queue) command (the default "default queue" is 0). If you do not make any other queues, every job goes through Q0. You can define from 1 to 100 job queues for your system. Each queue must have a unique number, between 0 and 1023.

For a discussion of the things you should consider if you want to make more queues, and for a list of the job queue attributes, refer to the *System Administration Guide*.

For a complete description of all job queue attributes, see the MQ (Make Queue) system command in the *System Commands Reference Manual.* For more information on the SQ (Show Queue), DQ (Default Queue), and QF (Queue Factors) system commands, see the same manual.

Creating, Modifying, and Deleting Job Queues (MQ)

Use the MQ (Make or Modify Queue) system command to create, modify, and remove job queues. The following procedures show you the basic command syntax. For a complete description of the MQ command and all the default values for the attributes, see the *System Commands Reference Manual.*

There are many system administration issues involved in creating and modifying queues. Talk to your system administrator or refer to the *System Administration Guide* before altering your system's queue structure.

Procedure for Creating Queues

You can create a job queue using all the default attribute values with the following syntax:

```
MQ <number>
```

This command creates a job queue with the number you enter. That number must be in the range 0 through 1023. The queue number only identifies the queue, and has nothing to do with priority.

To create a queue with special attributes, use this general syntax:

```
MQ <number> <queue attribute>
```

A full explanation of every job queue attribute is in the *System Commands Reference Manual.*

If you do not specify a value for every queue attribute, only the attributes you specify are initialized. The others get the system default values.

Procedure for Modifying Queues

To modify a job queue, use this basic syntax:

```
MQ <number> <queue attribute>
```

The number is the number of the existing queue you wish to modify.

Look in the *System Commands Reference Manual* for a complete explanation of all the job queue attributes.

Only the attributes you include in the command are changed. All other attributes are unaffected.

Note that changing certain attributes can cause all or some of the jobs in the changed queue to be queue-discontinued (Q-DS).

For example, some jobs might have estimates of system resource requirements associated with them. If the PROCESSTIME queue attribute has a value, then any job with a higher estimated value for process time than the job queue attribute value is not allowed in the queue.

So, if you lower the value of the PROCESSTIME attribute in a queue, the system takes any jobs with a higher estimate value out of the queue, or discontinues them.

Procedure for Deleting Queues

To remove a job queue, enter the following:

```
MQ- <number>
```

The number is the identification number of the queue. If there are no active jobs in the queue, it is eliminated immediately.

If there are active entries in the queue, you get the following message:

```
QUEUE HAS ACTIVE JOBS
```

In this case, the queue is not removed.

Jobs waiting in the queue must be moved to other queues, or they are discontinued.

Displaying Job Queues and Queue Attributes (SQ, QF)

Use the SQ (Show Queue) system command either to display all the queues defined on the system and the first job in each queue, or to display all the jobs in a single queue.

The QF (Queue Factors) system command displays all the queue attributes and their values for every job queue at once, or for just a single job queue.

Procedures for Displaying Job Queues

- To display all the job queues on your system and the first job in each queue (for example, the job with the highest priority), enter

  ```
  SQ
  ```

- To display all the jobs in a single queue, enter

  ```
  SQ <number>
  ```

  The number is the number of the job queue you want to look at.

Procedures for Displaying Queue Attributes

- To display the value of all the attributes for each job queue on your system, enter

  ```
  QF
  ```

- To display the values of the attributes for a single job queue, enter

  ```
  QF <number>
  ```

  The number is the number of the queue you want to examine.

For complete descriptions and default values of the job queue attributes, see the MQ (Make or Modify Queue) system command entry in the *System Commands Reference Manual.*

Examples

- The following is an example of an SQ command display. The example system has four queues defined: 0, 2, 4, and 5

  ```
  QUEUE 5
   NO ENTRIES
  QUEUE 4 (FIRST OF 3 ENTRIES)
   5645 55 (OPERATIONS)DAILY/BACKUP
          QUEUED: Ø4/15/92 AT 8:3Ø:42   STARTTIME = 19:3Ø:ØØ
  QUEUE 2
   NO ENTRIES
  QUEUE Ø
   NO ENTRIES
  ```

If a queue has entries, the SQ command displays the first entry in the queue, as well as the number of jobs queued. The first line of the job entry display includes, from left to right:

- The job number
- The job priority
- The usercode the job was started under, if any
- The job name

The second display line of the job entry includes, from left to right:

- The date the job was entered in the queue
- The time the job was entered in the queue
- The time the job is scheduled to start

- The following is an example of an SQ display when the command is used to display a specific queue. The command for the example is *SQ 5*:

```
QUEUE 5
5645 55 DAILY/BACKUP
        QUEUED: 4/15/92 at 8:30:42   STARTTIME = 19:30:00
5634 50 SQUASH
        QUEUED: 4/15/92 AT 8:31:27   STARTTIME = 2:00:00 ON 4/16/92
5686 50 (JESTON)STARTUP
        QUEUED: 4/15/92 AT 8:35:04   STARTTIME = 7:00:00 ON 4/16/92
```

The display contents are described in the preceding example.

- The following is an example of a QF command display:

```
QUEUE 0:
  DEFAULT QUEUE
  MIXLIMIT = 8
  ACTIVE COUNT = 2
  DEFAULTS:
    PRIORITY = 50
  LIMITS:
    PRIORITY = 99
QUEUE 5:
  MIXLIMIT = 15
  ACTIVE COUNT = 1
  DEFAULTS:
    PRIORITY = 50
  LIMITS:
    NONE
QUEUE 16:
  MIXLIMIT = 2
  DEFAULTS:
    PRIORITY = 45
  LIMITS:
    PRIORITY = 99
```

Moving Jobs within a Job Queue (MOVE)

Use the MOVE (Move Job/Pack) system command to change the order of jobs in a job queue. This is useful if you need one job to execute immediately or before other jobs.

Changes made to a job queue with the MOVE command are temporary. They do not last over halt/loads. If the system is halt/loaded, the job queue reverts to its original order.

Procedure

To move a job within a job queue, use the following syntax:

```
MOVE <mix number list> : <mix number>
```

This command places the job with the mix number following the colon (:) before the jobs in the mix number list. All the jobs referred to with this command must be in the same job queue.

Example

Suppose you have the following job queue:

```
QUEUE 5
 1510 55 DAILY/BACKUP
         QUEUED: 08/10/92 AT 7:04:50   STARTTIME = 19:00:00
 1344 50 JOB/LOGOUT/SMFII
         QUEUED: 08/10/92 AT 0:08:57   STARTTIME = 19:00:00 ON 08/11/92
 1507 50 DAILYUPDATED
         QUEUED: 08/10/92 AT 7:04:44   STARTTIME = 19:00:00
```

The leftmost field is the mix number (the number to the right of that is the job priority). If you want to move DAILYUPDATED in front of JOB/LOGOUT/SMFII, you enter

```
MOVE 1344 : 1507
```

It helps if you think of it as moving 1344 after 1507. Read the colon as *after*, for example, *MOVE 1344 after 1507*.

So if you want to move DAILYUPDATED to the top of the queue (move DAILY/BACKUP and JOB/LOGOUT/SMFII after DAILYUPDATED), you enter

```
MOVE 1510, 1344 : 1507
```

If you enter this command, and then display the example queue with the SQ (Show Queue) system command, it looks like this:

```
QUEUE 5
 1507 50 DAILYUPDATED
         QUEUED: 08/10/92 AT 7:04:44   STARTTIME = 19:00:00
 1510 55 DAILY/BACKUP
         QUEUED: 08/10/92 AT 7:04:50   STARTTIME = 19:00:00
 1344 50 JOB/LOGOUT/SMFII
         QUEUED: 08/10/92 AT 0:08:57   STARTTIME = 23:59:00 ON 08/11/92
```

Scheduling Start Times for Jobs in Job Queues (STARTTIME)

You can assign a start time and date to jobs in a queue with the STARTTIME (Start Time) system command.

This command automatically prevents the job from starting until the time and date you specify. Your job starts processing at the time you specify as soon as system resources are available to run your job. Scheduling start times is helpful, for example, for scheduling big jobs to run after normal business hours.

You can display the start times, if any, of the jobs in a queue with the SQ (Show Queue) system command described earlier in this subsection.

Procedure

To assign a start time for the present date, enter

```
<mix number list> STARTTIME = <time>
```

The mix number is the mix number of the job you want to assign the start time to. You can assign a start time to more than one job by separating mix numbers with a comma. The time is in the 24-hour format.

To assign a start time and date, enter

```
<mix number list> STARTTIME = <time> ON <date>
```

The syntax is the same as in the previous syntax, adding the ON <date>, where the date is in the form *mm/dd/yy*.

Example

To assign a start time of 4:45 on 5/19/92 to the job with a mix number of 7481, enter

```
7481 STARTTIME = 16:45 ON Ø5/19/92
```

Keys Files, Installing (IK)

When you receive new software release tapes, or purchase new software for your system, the tapes you get are called *conditioned tapes*. Files on conditioned tapes are guarded by "keys."

To copy one of these guarded files to your system, you must have a key for that file on your system, or you cannot copy it. You get keys onto your system by installing a keys file.

A keys file is sent to you with every software shipment on a standard library maintenance tape. The name of the tape for a Mark release is KEYSTAPE. The name of the keys file on that tape is SYSTEM/<release ID>/KEYSFILE. This keys file contains the keys for the software you are licensed for. You need to install these keys on your system so that you can copy the software you have purchased.

Normally, you install the keys file when you install the software with the INSTALL (Install Software) system command. You can, however, install the keys file separately with the IK (Install Keysfile) system command.

For information on the INSTALL command, and for information about installing software on your system, see the *Software Release Installation Guide*.

Procedure

Use the following procedure to separately install new keys files.

1. Copy the keys tape to a disk on your system. You can name it anything you like (except *SYSTEM/KEYSFILE).

   If you need to know how to copy a file from tape, refer to "Copying Files (COPY, ADD)" in this section.

2. Use the MERGE option of the IK command to merge the new keys file with the keys file already on the halt/load family. This creates a new version of *SYSTEM/KEYSFILE and removes the old one. Enter

   ```
   IK MERGE <file name> ON <family name>
   ```

   The file name is the name of the new keys file you copied to your system, and the family name is the name of the disk you copied it to. For example:

   ```
   IK MERGE NEW/KEYSFILE ON USERDISK
   ```

   This command installs the file NEW/KEYSFILE. USERDISK is the disk the new keys file was copied to. The command merges the new file with the keys file on your halt/load family and removes the old version.

   When you enter the IK MERGE command, the new keys file is installed.

You can look at the contents of the updated keys file with the SHOW option of the IK command:

```
IK SHOW
```

Libraries, Displaying (LIBS)

Use the LIBS (Library Task Entries) system command to display all the frozen libraries on your system.

A library can be a program, but it is usually a collection of routines stored in the same file. These routines are used by other programs, so that each program does not have to repeat code for common routines.

Libraries normally leave the mix if no other programs are currently accessing them and they are not permanent. Permanent libraries stay in the mix all the time, even if no programs are currently using them.

This command is useful for verifying that a support library for a program exists. An example of this is in the task "Switching DATACOMINFO Files" in this section.

Since some libraries are in the mix all the time, your supervisor might want to "suppress" them. Suppressing a library simply prevents it from showing up in the normal displays, so it does not take up space in those displays. Following are procedures for displaying both suppressed and unsuppressed libraries.

For more information about the LIBS command and an explanation of its displays, see the *System Commands Reference Manual.*

Procedures

- To display all unsuppressed frozen libraries, enter

  ```
  LIBS
  ```

- To display all frozen libraries, including suppressed libraries, enter

  ```
  LIBS ALL
  ```

- To display only the libraries running with a specific usercode, enter

  ```
  LIBS USER <usercode>
  ```

Libraries, Support (SL)

The SL (Support Library) command maps function names to library code files. You can use the SL system command to do the following:

- Display current support library assignments.
- Assign a function name to a library code file.
- Change a current support library assignment to another library code file.

The SL command is used to reference the support libraries. Because the system controls the function-mapping tables for library linkage, your programs can access a library through the appropriate function name.

When you use the SL system command to assign a function name to an existing library code file, all programs that call that function are directed to that specific library file for as long as that linkage remains in force.

The SL system command enables you to change to new libraries without affecting any running programs or requiring a new library name to be compiled into the calling program.

You already have several function names and associated libraries that are provided in your system software. For example, many intrinsics reside in the SYSTEM/GENERALSUPPORT library, which has the function name GENERALSUPPORT. All function names suffixed with SUPPORT are reserved for current and future system use. However, you can change the titles of the support libraries or create function names through the SL command.

Procedures

- To display the support libraries associated with their function names, enter

  ```
  SL
  ```

- To assign a function specification to the system tables, enter

  ```
  SL <function>
  ```

- To initialize a function, or to change the specification of a function that is already defined, enter

  ```
  SL <function> = <file title>
  ```

  If a function specification is changed to a different library, tasks already linked to the old library are not affected. New linkage requests for tasks use the new library. The exception to this is if the old library was listed as a *ONEONLY* library and had been initiated or established with an SL command since the previous halt/load.

  If a halt/load occurs after a function specification is changed, all tasks restart using the new library.

- To remove a function specification from the system tables, enter

```
SL- <function>
```

When a function specification is removed, all running tasks continue to use the old library. For new tasks, the system creates the following waiting entry message:

```
FUNCTION <function name> IS NOT DEFINED.  DEFINE WITH SL OR DS.
```

Examples

To display the current assigned support libraries on your system, enter

```
SL
```

The system displays a response similar to the following in the system messages area:

```
SL GENERALSUPPORT = *SYSTEM/GENERALSUPPORT
SL PLISUPPORT     = *SYSTEM/PLISUPPORT
SL PRINTSUPPORT   = *SYSTEM/PRINT/SUPPORT
                    *SYSTEM/PRINT/SUPPORT/2 (PENDING)
SL USERFUNCTION   = (USER)SYSTEM/USERLIBRARY
```

In the example above, all system files are marked with an asterisk (*). PRINTSUPPORT appears to have two printsupport files. The file marked (pending) is not currently active, but has been assigned to be the PRINTSUPPORT file after the next halt/load. See below for more information about PRINTSUPPORT files.

To inquire about the library code file assigned to GENERALSUPPORT, enter

```
SL GENERALSUPPORT
```

The system displays a response similar to the following in the system messages area:

```
SL GENERALSUPPORT = *SYSTEM/GENERALSUPPORT
```

This message indicates that the system library file *SYSTEM/GENERALSUPPORT has been assigned to the GENERALSUPPORT function.

To assign a library file with the usercode (USER) to the function USERFUNCTION, enter

```
SL USERFUNCTION = (USER)SYSTEM/USERLIBRARY
```

The system responds with a message similar to the following:

```
FUNCTION "USERFUNCTION = (USER)SYSTEM/USERLIBRARY" ESTABLISHED
```

For *ONEONLY* support libraries, such as PRINTSUPPORT, the title cannot be changed while the library is in use. In that case, the new title is appended under the function name in the SL function mapping table and the following message is displayed:

```
FUNCTION 'XXX = YYY' WILL BE ESTABLISHED
```

The new title replaces the current one when the library is no longer in use or when a halt/load takes place. The pending title can be removed by the *SL–* command.

For more information about system library attributes, refer to the *System Commands Reference Manual*, the *A Series GETSTATUS/SETSTATUS Programming Reference Manual*, and the *Security Administration Guide*.

MARC, Transferring to and from (??MARC)

You can use Menu-Assisted Resource Control (MARC) from the system control terminal. The MARC menus and help text are useful in guiding you through complex or unfamiliar tasks.

For MARC documentation, see the *MARC Operations Guide*.

Procedure

To get into MARC, do the following:

1. Transfer to MARC by entering

   ```
   ??MARC
   ```

   If you see the MARC log-on screen, skip to step 3.
2. Transmit a blank (press the space bar, then the XMIT key).

   This brings up the MARC log-on screen.
3. Enter the usercode and password.

   Obtain the Operations usercode and password from your system administrator.

Once you are through in MARC, do the following to get back to ODT mode:

1. Enter BYE on the MARC home menu Choice line.

   You should then see the MARC log-on screen again.
2. Enter the following on the Usercode line:

   ```
   ??ODT
   ```

   You should now be back in ODT mode.

Master Control Program (MCP)

The master control program (MCP) is the operating system that controls your computer. It is just a big program, like any other program (although the MCP does have special qualities—you cannot discontinue or stop it, for example). The term *MCP* as used in this guide means the same thing as *operating system*.

For a more complete explanation of the MCP and its functions, see "System Software" in Section 1.

This subsection describes three tasks dealing directly with the MCP: changing the MCP your system is running on, determining what the current MCP is, and changing MCP system options.

Changing MCPs (??CM)

The MCP is the program that controls your system, and you can change the MCP your system is running on. This includes updating to a new MCP, falling back to an older MCP, using an MCP with special features (such as "diagnostics"), and simply switching to another copy of the MCP you are currently using if it has become corrupted somehow.

Since the MCP is an important program on your system, there are some issues you need to think about.

If you are installing a new MCP from a software release, refer to the *A Series Software Release Installation Guide*. It contains complete, detailed instructions on installing and testing new MCPs.

If you are not sure about the MCP you are changing to, you can use a form of the ??CM command that makes the MCP change temporary. This form of the command is discussed in the following procedure.

Procedure

Use the following procedure to change the MCP your system is running on. Note that this procedure causes a halt/load.

1. Copy the MCP code file you are changing to onto the halt/load family if it is not already there.

   If you do not know how to copy a file, refer to "Copying Files (COPY, ADD)" in this section.

   If you do not know what the MCP code file name or the halt/load family is, enter the WM (What MCP) system command. This shows the code file name on the first line, and the halt/load unit number on the second line of the display.

2. Enter one of the following ??CM (Change MCP) system commands:

   Note: *The ??CM system command causes an immediate halt/load.*

   - To temporarily change the MCP, enter

     ```
     ??CM# <file name>
     ```

     This form of the command temporarily changes the MCP to the code file with the specified file name. The system changes back to the original MCP at the next halt/load. Note that you must include the pound sign (#) for change to be temporary.

     This command is useful if you are unsure of the MCP you are changing to.

   - To change the MCP permanently, enter

     ```
     ??CM <file name>
     ```

     This form of the command changes the MCP to the code file with the specified file name until you change it again with another CM command. The change lasts across halt/loads.

The preceding procedure uses the primitive form of the CM (Change MCP) system command, *??CM*. The primitive form of the CM command works exactly like CM, except that it changes the MCP immediately. The nonprimitive version waits for a null mix (no active, waiting, or library entries in the mix). The nonprimitive version is often not practical because there are so many libraries that remain in the mix all the time.

Displaying the Current MCP (WM)

Use the WM (What MCP) system command to display the name of the MCP code file, and information about the software version of the MCP your system is currently running on.

Procedure

To display the current MCP name and information, enter

```
WM
```

Example

This is an example of a WM display from an A 15:

```
MCP: *SYSTEM/ASD/MCP/40086G 40.086.1965
H/L UNIT: 96
COMPILED: 03/27/92 @ 10:39:11 (NEWP 40.85)
COMPILE TIME OPTIONS ARE:
       TRACE               DIAGNOSTICS         EXPERIMENTAL
       LINEINFO            LOCKTRACE           READLOCK
       READLOCKTIMEOUT     RESTART             ASDDEBUG
H/L REASON: CM REQUEST
H/L TIME: Saturday March 28, 1992 (92088) 5:52 PM.
GROUP ID: MPA15C
HOSTNAME: MPA15C
SYSTEM SERIAL NO: 8
SYSTEM TYPE: A15
CATALOG LEVEL: 0
NEXT MCP: NOT SPECIFIED
The MCP bound LANGUAGE is :    ENGLISH
CONTROLWARE TYPES SAVED:  B9387 B9389 B9387S
```

Setting and Resetting MCP Options (OP)

The master control program (MCP) has 44 available options for tailoring your operating environment. The *System Commands Reference Manual* has complete descriptions of all the MCP options.

The following is a list of all the available MCP options and their numbers:

No.	Option	No.	Option
1.	OPEN	25.	not used
2.	TERMINATE	26.	LOGPOSITIONING
3.	NOCHECK	27.	SERIALNUMBER
4.	LPBDONLY	28.	ARCHIVING
5.	AUTORM	29.	not used
6.	DIAGNOSTICS	30.	LOCKTRACE
7.	CDONLY	31.	IORANGECHECK
8.	AUTORECOVERY	32.	not used
9.	DUPSUPERVISOR	33.	KEYEDIOII
10.	DUPINTRINSICS	34.	MIRRORING
11.	TRANSWARNINGS	35.	DIAGNOSTICDUMP
12.	AUTODC	36.	AUDIT
13.	NODUMP	37.	FILESATURATION
14.	CPBDONLY	38.	EOTSTATISTICS
15.	AUTORUNNING	39.	PATHBALANCING
16.	CRUNCH	40.	NETRECOVERY
17.	BACKUPBYJOBNR	41.	LOGIOERRORSNEW
18.	PDTODISK	42.	ISCDEBUG
19.	NOFETCH	43.	IODIAGNOSTICS
20.	RESOURCECHECK	44.	PORTDEBUG
21.	not used	45.	USECATDEFAULT
22.	DIRDEBUG	46.	CATTEST
23.	CATALOGING	47.	MCPTEST
24.	OKTIMEANDDATE		

The system administrator for each site determines which of the MCP options are set and which are normally reset. For a discussion of the administration considerations dealing with MCP options, see the *System Administration Guide.*

For more information about the OP command and explanations of all the MCP options, see the *System Commands Reference Manual.*

Procedures

- To list the options that are set on your system, enter

  ```
  OP+
  ```

- To list the reset options, enter

  ```
  OP-
  ```

- To see all the options and their respective settings, enter

  ```
  OP
  ```

- To set an MCP option, enter

  ```
  OP+ <option list>
  ```

  The option list can be one or more of the MCP options, separated by blanks. You can use either the option number or the option name.

- To reset an MCP option, enter

  ```
  OP- <option list>
  ```

  The option list is the same as that described for setting MCP options.

Example

If you want to set options 8, 12, and 27, you enter

```
OP+ AUTORECOVERY 12 SERIALNUMBER
```

Memory Dump, Requesting

Memory and program dumps are used to help diagnose software problems. A dump is a file of memory contents which can be printed or written to tape. You can request a dump for the entire system, for a set of programs, or just one program. The system or individual programs sometimes automatically request a system dump or a program dump.

Only the basics of the dump commands are covered here. For detailed information on these commands, including a description of all the dump options, see the *System Commands Reference Manual.*

A dump of the entire system temporarily halts the processing for all jobs and tasks on the system, so make sure you have a good reason before requesting one.

The DUMP command can be used to create both memory dumps and program dumps. The DUMP command dumps the entire contents of system memory or the memory for one or more programs. It has an extensive option list enabling you to tailor the program dump to your needs.

The ??MEMDP primitive command also dumps the entire contents of system memory or the memory for a program. It does not have the program dump options DUMP has, and you can dump the memory for only one program at a time. It is valuable for forcing dumps when the system is not operating properly, and when the ODT is not responding to normal system commands.

You can request a program dump when you discontinue a job or task by including a dump option specification in the DS (discontinue) system command. Doing so is valuable if you discontinue a job or task because it is not running properly. The dump option list for the DS command is discussed in the *System Commands Reference Manual.*

Whether a program dump goes to a printer or to disk is determined by the MCP option PDTODISK. The default for this option is for dumps to be sent to a printer, but this option might have been changed. You can find out what the option is set to with the OP (Option) system command, which is described in the "Master Control Program (MCP)" subsection. Note that dump options specifying where a dump is sent in a DUMP or DS command override the MCP option.

Note:

- *If you are an Entry and Medium Systems (EMS) user and you do not want the data of your dumps compressed, you must set the UNCOMPRESSED option of the MDT (Memory Dump Type) system command. Compressing the data on EMS systems can considerably increase the time needed to copy a dump. This delay can cause problems if your system is part of a network. This condition is not true on A 12 systems through A 19 systems.*

  You can set the MDT UNCOMPRESSED *system command option by entering the following system command at the operator display terminal:*

  ```
  MDT UNCOMPRESSED
  ```

- *All data in subsequent system dumps is uncompressed until you reset the MDT* UNCOMPRESSED *command option. See the* System Commands Reference Manual *for an explanation of the MDT (Memory Dump Tape) system command and its options.*

Memory Dumps and Program Dumps

You should be aware that memory dumps differ significantly from program dumps, as follows:

- While a memory dump is in process, all other system processing stops.
- Memory dumps require special processing to print the analyzed information.
- Memory dumps are often generated automatically.

Program dumps have the following characteristics:

- While in process, program dumps do not disrupt system processing.
- Program dumps do not need special processing by programs such as SYSTEM/DUMPANALYZER to make them readable.
- Program dumps can be forced at the time an operator decides to discontinue a running job.

Procedures for Using DUMP

The DUMP system command can be used to create both memory and program dumps.

- To dump all of system memory, enter

  ```
  DUMP
  ```

  You can enter the reason for the memory dump into the dump listing with this syntax:

```
DUMP <text>
```

Only the first 16 characters are printed in the dump listing.

- To get a dump for one or more programs, enter

```
<mix number list> DUMP
```

The mix number list is the mix number or mix numbers for the program or programs you want a dump for.

There are a number of dump options you can choose when requesting a program dump with this command. All are explained in the *System Commands Reference Manual.*

If you are still not able to force the system to dump because of some severe system malfunction, consult the Unisys customer engineer assigned to your installation to resolve the problem.

Procedures for Using ??MEMDP

- To dump all system memory, enter

```
??MEMDP
```

You can enter the reason for the memory dump in the dump listing with this syntax:

```
??MEMDP <text>
```

Only the first 16 characters of the text show up in the dump listing.

- To dump the memory for a program, enter

```
?? <mix number> MEMDP
```

The mix number is the mix number of the program you want the memory dump for.

Procedure for Requesting a Program Dump with the DS (Discontinue) Command

You can get a program dump for a job or task when you discontinue it. To do so, enter

```
<mix number> DS ALL
```

The word ALL in the command is a DS command dump option, requesting a full dump for that job or task. You can specify other dump options as described in the *System Commands Reference Manual.*

Examples

To dump the program memory for the tasks with mix numbers 3132 and 3134, enter

```
3132, 3134 DUMP
```

You can also get memory dumps for these tasks when you discontinue them by entering

```
3132 DS ALL

3134 DS ALL
```

Dump to Disk Capabilities

Some types of hardware or software malfunctions do not respond to the ODT dump procedures. To capture dumps caused by these malfunctions, you need to set up your system to automatically dump memory contents to a specified disk file by following the procedures described in this section.

When a memory dump is taken to a disk file, one of the two following dumpdisk files is used:

1. The DN file, a file that you specially designate by the DN (Dump Name) system command. The system attempts to dump memory contents to this file first.
2. SYSTEM/HLDUMPDISK, a special dumpdisk file established by the HLDUMPDISK option of the CM system command.

Sometimes memory dumps occur early in the initialization process. If a dump occurs before any disk families have been verified, the operating system cannot locate the designated DN file. In this case, if you have set the HLDUMPDISK option, the system proceeds to write a dump directly to the HLDUMPDISK file.

If both a DN file and a HLDUMPDISK file exist, and the dump does not occur early in the initialization process, the operating system tries to write the memory dump to the DN file. If the DN file is full, the operating system then attempts to write the memory dump to the HLDUMPDISK file.

Dumpdisk File Characteristics

A dumpdisk file is simply a large data file that has a rowsize of 730 segments (or multiples of 730). A dumpdisk file that is in use is marked as a SYSTEMFILE to prevent its removal. Loaded dumpdisk files are always opened EXCLUSIVE by the MCP to prevent any unauthorized access to or tampering with their data.

A dumpdisk file (either DN or HLDUMPDISK) can contain up to 13 separate memory dumps at one time.

Memory Dumps to the DN File

Any file you specify in a DN (Dump Name) system command can be used for memory dumps. The files specified in the DN command can reside on any family of a system, and can have rows on any member of its family.

After each halt/load, the system attempts to reload the DN file as an in-use dumpdisk file. If this attempt fails for any reason (such as that the file or family is no longer present, the system is unable to access all the rows, or the file is present but no longer a dumpdisk file), the DN specification is ignored.

Memory Dumps to the HLDUMPDISK File

In addition to the DN files, you can use the HLDUMPDISK file for memory dumps. HLDUMPDISK is a special dumpdisk file that resides entirely on the halt/load unit of a system. The name of the HLDUMPDISK file is always in the form SYSTEM/HLDUMPDISK/nnn where nnn is the family index on which the HLDUMPDISK file resides completely.

An HLDUMPDISK file is established, changed in size, or removed from system use by the CM (Change MCP) system command. You use the CM system command with the HLDUMPDISK option to designate a disk as a halt/load unit with HLDUMPDISK.

An HLDUMPDISK file always has a physical rowsize of 730 segments.

Refer to the *System Commands Reference Manual* for more detailed descriptions of the DN and CM commands.

Setting Up Memory Dump Files on Disk

The HLDUMPDISK files are loaded with the HLDUMPDISK option of the CM system command. Refer to the *System Commands Reference Manual* for more information on the CM command.

Procedure to Establish an Area for Memory Dumps on the Halt/Load Pack

Use the following procedure to establish an area for memory dumps on the halt/load pack:

1. To direct the system to write memory dumps on the halt/load disk, enter

```
CM +HLDUMPDISK
```

The system displays the following in the waiting entries:

```
-----------Job-Task-Pri---Elapsed-----------1 WAITING ENTRY------------
          2357/2357   8Ø      1:3Ø  JOB CHANGEMCP
               ACCEPT:ENTER NUMBER OF UNITS OF 16K WORDS(MINIMUM 128)
```

2. Answer the waiting entry request by entering the AX command and the minimum number of units, 128.

The system searches for an existing usable SYSTEM/HLDUMPDISK file.

- If one exists, the system uses that file.
- If no such file exists and the appropriate amount of space is available, the system creates a SYSTEM/HLDUMPDISK file.

- If there is not enough space on the halt/load disk, a message comparable to the following is displayed:

```
-----------Job-Task-Pri---Elapsed-----------1 WAITING ENTRY------------
         2357/2357   8Ø      1:3Ø  JOB CHANGEMCP
              PK44 73Ø SECTORS REQUIRED ON DISK
```

  Use the DS command to cancel the CHANGEMCP task.

If you have additional disk packs, you can place the memory dump file on one of those packs. To accomplish this task, follow this procedure:

1. Use the DN (Disk Name) system command to create and assign the memory dump file. Check with your system administrator before you decide where to assign the memory dump file and whether you want to compress the memory dump. The way you decide to store and retrieve a memory dump can adversely affect your system's performance. Refer to the explanations of the DN and MDT system commands in the *System Commands Reference Manual.*

   The following command creates a file with the name of MEMORY/FILE on a disk named SCSI1:

```
DN MEMORY/FILE ON SCSI1
```

   The system displays the following message in the waiting entries:

```
-----------Job-Task-Pri---Elapsed-----------1 WAITING ENTRY------------
         9776/9776   8Ø      1:3Ø  JOB DUMPDISKMASTER
              ACCEPT:ENTER NUMBER OF UNITS OF 16K WORDS(MINIMUM 128)
```

2. Answer the waiting entry request by entering the AX command and the minimum number of units, 128.

Use the DN (Disk Name) system command alone to inquire about currently assigned dump files:

```
DN
```

The system displays a message like the following in the message area:

```
DN-ED FILE :    *MEMORY/FILE ON SCSI1
K SECTORS        TOTAL/FREE
DN-ED FILE       793/793
DEFAULT DISKDUMP DISPOSAL (DL DPFILES) CREATE MEMORY/DUMP TAPES
```

You can find more information about the DN command in the *System Commands Reference Manual.*

How the MCP Uses Dumpdisk Files

Whenever possible, the MCP attempts to take a memory dump to a dumpdisk file rather than to tape. For disk, memory dumps are automatically executed to the DN file first, if possible; otherwise, the dump is executed to HLDUMPDISK, if possible, without operator intervention.

If the system cannot direct a memory dump to disk without overwriting dump information already in a dumpdisk file (destroying one or more dumps previously taken to the file), the system searches for an online scratch tape with the serial number DUMMMP. If it finds such a tape it writes the memory dump to that tape. Otherwise, you are requested to respond to the system dump procedure. You can approve the file overwrite (specifying which dumpdisk to use if both the DN file and the HLDUMPDISK file are available), or you can supply the number of the tape unit where you have mounted your dump tape. For example, you want to direct a memory dump to the tape unit MT15. The system notifies you that a dump is in progress, and prompts you to enter the number of the tape unit to be used. You enter

```
MT15
```

The system writes the memory dump to the tape you mounted on tape unit MT15.

After the memory dump is completed, the operating system attempts to empty the dump from the file as quickly as possible so the dumpdisk space is again available for use. You normally empty a dump from a dumpdisk by converting it to either a MEMORY/DUMP tape, or a disk file with the title DP/<date>/<time>/<dump reason>. A DP file is exactly like a raw MEMORY/DUMP tape. When SYSTEM/DUMPANALYZER is run, its file TAPEIN can be equated to the proper DP file.

Using the DUMPDISKMASTER Procedure

The DUMPDISKMASTER procedure handles the emptying of dumpdisk files after a dump. The DUMPDISKMASTER procedure is automatically initiated as an independent runner. You can initiate the DUMPDISKMASTER procedure if it is not already running by using the DF (Empty Dumpdisk File) system command. You can initiate the DUMPDISKMASTER procedure to empty all the dumpdisk files of the system, or to empty a particular dumpdisk file. The specified file might not be currently loaded as the DN or HLDUMPDISK file. Only one DUMPDISKMASTER procedure can be working at a particular time. Refer to the *System Commands Reference Manual* for a description of the DF system command.

Procedure for Responding to the DUMPDISKMASTER RSVP Message

For every dump taken, the DUMPDISKMASTER procedure issues an RSVP message specifying the date, time, and reason for the dump in question. Respond with

```
<mix number> AX <action>
```

The mix number is the mix number of the DUMPDISKMASTER procedure. The action is any one of the following:

Action	Description
OK *TAPE	Converts the dump to a tape file.
OK <family name>	Converts the dump to a disk file.

continued

continued

Action	Description
PURGE	Discards the dump with no further processing.
SKIP	Postpones the final disposition of the dump. The SKIP response does not free any diskdump file space for use, and normally should not be used.
OK	Creates a DP file on the family specified by the DPFILES option in the DL (Disk Location) system command. The dump is emptied into this file. Refer to the *System Commands Reference Manual* for a description of the DL system command.

If DL DPFILES is null (it is null until set otherwise, and can be nulled by setting it to a period (.) with a DL system command), the OK action causes a diskdump to be converted to a MEMORY/DUMP tape.

Once running, the DUMPDISKMASTER procedure does not terminate until a disposition has been made of each diskdump on each dumpdisk file in use by a system, or until you discontinue it.

The DUMPDISKMASTER procedure also handles DN specification changes and the initialization of dumpdisks after a halt/load. After a halt/load has occurred, the DUMPDISKMASTER procedure checks whether each dumpdisk of the system is large enough to hold a single dump of current memory. If one file is not large enough, the DUMPDISKMASTER procedure attempts to increase the dumpdisk's capacity to match current memory. If there is insufficient disk space available to accomplish this, a special dumpdisk warning is issued. This warning stays in the waiting entries of the system until you enter an OK action or discontinue the process.

Memory Usage, Displaying (CU)

Use the CU (Core Usage) system command to display information about the memory usage either for a particular job or task, or for the entire system.

This command display contains information about how much available memory is left on the system at the time the command is entered. This display can be helpful if your system has a slow response time and you are looking for reasons why, or if the OTHERPBIT value for the system is high and you are trying to determine why.

Procedures

- To display memory usage information for the entire system, enter

```
CU
```

- To display memory usage information for a job or task, enter

```
<mix number> CU
```

Examples

The following is an example CU display:

```
------------------------------- SYSTEM MEMORY USAGE -------------------

 --- Memory Summary --    -- ASD Information --    ----- Buffer Statistics
 Available      1219789   Total ASDs      209714   Size source   Number
 Non save       5353791   In use             42%   Buffergoal         0
 Save           3912180   Maximum in use     44%   Buffersize         0
 Total         10485760   Factor    50 Words/ASD   Blocksize        754
```

AVAILABLE refers to the number of words of memory available for executing jobs and tasks. NON SAVE refers to memory that can be overlaid (written to disk and then made available), and SAVE refers to memory that cannot be overlaid.

This is a CU display for job 3132:

```
3132 MEMORY:      TOTAL        SAVE
  STACK           13378        8888
  CODE            21962         445
    TOTAL:        35340        9333
```

Messages, Displaying (MSG)

Use the MSG (Display Messages) system command to display the most recent messages from the system.

Although a system messages area is usually included in the automatic display mode (ADM) display, due to space limits, you can see only about 7 messages in this display.

The MSG command enables you to see the 21 most recent system messages. The most recent messages are displayed at the top of the screen.

This command is useful if new messages overwrite the messages that you want to see on the ADM display entry.

For more information about the MSG command and display, see "Monitoring Processes at the ODT" in Section 3, or see the *System Commands Reference Manual.*

Procedure

To display messages, enter the following:

```
MSG
```

To keep this display from being replaced by the ADM displays, press the Local key. Use the NS (Next Screen) system command to view more than the first screen of messages.

Messages, Suppressing (MSC)

If you are a user with SYSTEMUSER status, you can use the MSC (Message Control) MARC command to enable the message suppression feature for your system. This feature enables you to simplify ODT and remote terminal displays by suppressing such repetitious messages as library maintenance copy messages and messages from specific programs.

The following text assumes that you have a good knowledge of MARC and its capabilities. If you need more information about MARC, refer to the *MARC Operations Guide*. In addition, refer to that guide for more in-depth information about the MSC command.

Note: *The message suppression feature enables you to suppress messages that are meant to warn the operator of system errors or potentially harmful situations. If you are a new or inexperienced operator, consult with your system administrator before you decide to limit the number of messages appearing on ODTs or terminals.*

The message suppression feature enables you to dynamically suppress or display messages

- By message type
- By a unique characteristic, such as charge code, task name, or usercode

In addition, you can control if messages are suppressed or displayed

- At the ODT
- At a remote terminal or terminals
- At both the ODT and a remote terminal or terminals

You can use the message suppression feature in any of the following ways:

- By entering MSC commands on the Action line of any MARC screen
- By selecting options on MARC screens
- By loading one or more files of MSC commands

  You can direct a file of MSC commands to be loaded automatically after a halt/load, or you can load MSC command files manually as needed. For more information about creating and loading files of MSC commands, refer to the *MARC Operations Guide*.

Messages that are suppressed on either ODT or remote terminals do not disappear completely. The messages still appear in printed job summaries. You can also display suppressed messages at the ODT by entering the system command

```
MSG ALL
```

Understanding How the System Uses MSC Commands

When the message suppression feature is inactive, the system displays the system messages as follows:

- Messages resulting from jobs initiated at the ODT appear on the ODT screen.
- Messages resulting from any jobs initiated at a remote terminal – that is, a terminal connected to the system by datacomm lines – appear at that terminal and at the ODT.

When the message suppression feature is active, the operating system reads the current list of MSC commands in working memory before it displays the message. If it finds an applicable message suppression or show command, the operating system either suppresses or shows the message at its destination, according to the last matching criterion the system encounters.

The operating system always executes MSC commands in sequential order. For this reason, an MSC command that suppresses all messages at the ODT can be overridden by a subsequent MSC command entered after it that directs all or selected messages to the ODT.

If you want to suppress or display only messages that come from programs that are initiated at remote terminals, be sure to include the REMOTE option of the commands.

Displaying Current MSC Commands

To display current active MSC commands, use the MSC form of the command on the Action line of any MARC screen or use the MSC selection on the MSC screen of MARC.

Procedure

On the Action line of any MARC screen, enter

```
MSC
```

If MSC commands are active on your system, a list of all MSC suppression and show commands is displayed on the MARC screen in the order they are processed. If no MSC commands are currently in use, a message appears on the MSC screen telling you that no MSC commands are currently active.

Controlling Messages of a Certain Type

The system software categorizes messages into types. You can use the messages types to suppress or display messages. The following table lists frequently used message types and the message ID that is associated with each category. Refer to this list when you suppress or display messages by message type.

Message ID	Type of Message
MSRDIR3	<file name> COPIED FROM <source> TO <destination>
MSRDIR4	<file name> REMOVED ON <family name>
MSRDIR7	<file name> CHANGED TO <file name>
MSRDIR10	<file name> NOT ON <family name>
MSRDIR54	<number> FILES REMOVED IN <directory name>
MSRDIR85	<file name> REPLACED ON <family name>
MSRDISP3	DISPLAY: <text>
MSRDPC97	<device> PURGED
MSRFIN9	NO FILE <file name>
MSRHI13	SEG ARRAY ERROR @ <line number>
MSRIPC1	MISSING CODE FILE @ <line number>
MSRPRQ19	<device> <file name> PRINTED AND REMOVED
MSRWARNG<number>	WARNING <number> <text of warning>

You can produce a detailed list of these report messages and message IDs for all message types recorded in the system log file by running the program SYSTEM/LOGANALYZER with the MSG option. For more information on the SYSTEM/LOGANALYZER program, see the *A Series System Software Support Reference Manual.* For more information about system messages, including their causes and your suggested responses, see the *A Series System Messages Support Reference Manual.*

Examples

- To request that file copy messages be suppressed at the ODT no matter where the programs that sent the messages are initiated, enter the following on the Action line of any MARC screen:

  ```
  MSC SUPPRESS MSG MSRDIR3 AT ODT
  ```

- To request that file copy messages be suppressed at the ODT only when the messages come from a program that is initiated at a remote terminal, enter the following on the Action line of any MARC screen:

  ```
  MSC SUPPRESS REMOTE MSG MSRDIR3 AT ODT
  ```

  Even though the copy messages are suppressed at the ODT, the messages continue to be sent to the remote terminal where the job originated.

If the command without the REMOTE option is already entered, the command with the REMOTE option is not necessary because all copy messages are already suppressed at the ODT.

Controlling Messages with a Unique Characteristic

You can also suppress or display messages that have a unique characteristic, such as a charge code, a task name, or a usercode.

Examples

- To request that messages be suppressed at the ODT that are sent from programs associated with a usercode—for example, OPS—no matter where the programs are initiated, enter the following on the Action line of any MARC screen:

  ```
  MSC SUPPRESS USER OPS AT ODT
  ```

- To request that messages be suppressed at the ODT that are sent from programs associated with a usercode—for example, OPS— that are initiated at a remote terminal, enter the following on the Action line of any MARC screen:

  ```
  MSC SUPPRESS REMOTE USER OPS AT ODT
  ```

  Even though the messages from programs associated with the OPS usercode are suppressed at the ODT, the messages continue to be sent to the remote terminal where the job originated.

  If the command without the REMOTE option is already entered, the command with the REMOTE option is not necessary because all messages from programs associated with OPS are already suppressed at the ODT.

Controlling All Messages

You can also suppress or display all messages.

Examples

- To request that all messages be suppressed at the ODT that come from programs initiated at remote terminals, enter the following on the Action line of any MARC screen:

  ```
  MSC SUPPRESS REMOTE ALL AT ODT
  ```

  The messages from jobs originating at remote terminals continue to appear at those remote terminals.

- To request that all messages be suppressed at the remote terminals where the programs are initiated, enter the following on the Action line of any MARC screen:

  ```
  MSC SUPPRESS REMOTE ALL AT MCS
  ```

Combining Commands for Unique Suppression

You can combine commands to accomplish suppression that is not possible with just one command. The following examples show just a few possible combinations.

Examples

- The following example demonstrates how to allow all messages from usercode OPS, including the messages with a message ID of MSRDIR3, to appear at both the ODT and remote terminals, while suppressing at the ODT messages with a message ID of MSRDIR3 that come from all other users at remote terminals:

  ```
  MSC SUPPRESS REMOTE MSG MSRDIR3 AT ODT
  MSC SHOW REMOTE USER OPS AT ODT
  ```

  The first command requests that all messages from programs that are initiated at remote terminals with a message ID of MSRDIR3 be suppressed at the ODT, but requests that each such message appear at the remote terminal where the job was initiated. This request does not suppress message display at the ODT for messages of the same type from jobs initiated at the ODT.

  The second command requests that all messages that are initiated by usercode OPS be displayed at the ODT when the programs are initiated at a remote terminal.

- The following example demonstrates how to suppress all messages from jobs run under usercode OPS at the ODT, regardless of where the jobs are initiated, and how to display all messages at the ODT from a job with the charge code 76583 even if the job was run under usercode OPS:

  ```
  MSC SUPPRESS USER OPS AT ODT
  MSC SHOW CHARGE 76583 AT ODT
  ```

  The first command requests that all messages from programs associated with usercode OPS be suppressed at the ODT no matter where the program is initiated.

  The second command requests that all messages from a job with the charge code 76583 be displayed at the ODT no matter where that job is initiated. Since this command follows the first command, messages from jobs run under charge code 76583 are displayed even if a job is associated with usercode OPS.

- The following example demonstrates how to display a "FILE NOT FOUND" message at the ODT no matter what usercode initiated the job, but at the same time, suppress messages from jobs with the BILLING usercode at the ODT. This example applies to jobs begun at any location.

  ```
  MSC SUPPRESS USER BILLING AT ODT
  MSC SHOW MSG MSRDIR1Ø AT ODT
  ```

  The first command requests that all messages from programs associated with the usercode BILLING be suppressed at the ODT no matter where the program is initiated.

  The second command requests that all "FILE NOT FOUND" messages be displayed at the ODT no matter where the programs that sent the messages are initiated. Since this command follows the first command, the "FILE NOT FOUND"messages from programs associated with usercode BILLING are displayed at the ODT.

- The following example demonstrates how to display all messages from programs associated with usercode SUE at the remote terminal where the programs are initiated, but suppress all file copy messages from any other programs that are initiated at remote terminals. In this example, neither of these commands affects the display of messages at the ODT.

  ```
  MSC SUPPRESS REMOTE MSG MSRDIR3 AT MCS
  MSC SHOW REMOTE USER SUE AT MCS
  ```

  The first command requests that all messages with the message ID of MSRDIR3 be suppressed at the remote terminals where the programs that sent the messages are initiated.

  The second command requests that all the messages from programs associated with usercode SUE be displayed at the remote terminal where the program is initiated.

Enabling and Disabling MSC Commands

The following forms of the MSC command are very important when you are using MSC commands:

- MSC SAVE <file title>

  The MSC SAVE command saves the current list of MSC commands in the designated file. The current list consists of all MSC commands that are currently active on the system.

- MSC CLEAR

  The MSC CLEAR command deletes all active MSC commands. When this form of the command is entered, the operating system reverts to default message suppression mode so that all messages are displayed at their intended destinations.

- MSC SUSPEND

  The MSC SUSPEND command temporarily suspends all active MSC message suppression commands. As a result, the message suppression feature reverts to default mode, in which all messages appear at all intended destinations. All message suppression commands are suspended until the MSC RESUME command is issued, or until the next halt/load occurs.

- MSC RESUME

  The MSC RESUME command causes message suppression to start again after the MSC SUSPEND command has been entered.

- MSC LOAD <file title>

  The MSC LOAD command appends a file of MSC commands to the currently active MSC commands on your system. To ensure that only the new MSC commands are active, perform the MSC CLEAR command before entering the MSC LOAD command. Manually loaded MSC commands remain active until the next halt/load.

This guide does not provide complete instructions for using the MSC LOAD command. You can find complete instructions for loading files of MSC commands, as well as information about creating your own file of MSC commands, in the *MARC Operations Guide*. That guide also explains how to create a file of MSC commands that is automatically loaded whenever a halt/load occurs.

Messages, Displaying Identity of Source (MP)

The MP (Mark Program) system command gives an identity to a program or library. This identity is then used to prefix any messages displayed by the program or library code file.

Displaying the program or library identity along with messages can be very useful when there are many messages being displayed at the ODT from many different sources. The identity prefix enables you to identify where a message came from, which can help in tracking down problems and in monitoring specific program status.

The identity that is used to prefix a message comes from the code file that performs the display. For example, if a program with an identity of PROG calls a library with an identity of LIB, any displays from the program are prefixed with PROG, and any displays from the library are prefixed with LIB.

Any messages which the operating system displays on behalf of a program or library are prefixed with the identity of the program or library. For example, if a program with an identity of ORDERS gets a "NO FILE" message, the message is prefixed with ORDERS.

Note that if the security administrator option has been enabled on your system, only users with the security administrator privilege are allowed to use the MP command. (You can check the value of this option by entering *??SECAD* at the ODT.)

Procedures

- To add an identity to a program or library, use the following syntax:

  ```
  MP <file title> + IDENTITY <identifier>
  ```

  The file title is the title of the program or library you want to add an identity to. The identifier is the string that will be prefixed to any messages from the program or library. Specifying an identifier for a program or library that already has one simply changes the identifier.

- To remove an identity from a program or library, use the following syntax:

  ```
  MP <file title> - IDENTITY
  ```

  The file title is the title of the program or library from which you want to remove the identifier.

Examples

Suppose you want to add the identity "PAYROLL" to the program *WEEKLY/PAYROLL, which is on the disk pack named ACCOUNTING. You enter

```
MP *WEEKLY/PAYROLL ON ACCOUNTING + IDENTITY PAYROLL
```

From that point on, any messages generated by the program are prefixed with PAYROLL.

Later, suppose you want to remove the identifier from the program. You enter

```
MP *WEEKLY/PAYROLL ON ACCOUNTING - IDENTITY
```

The program no longer has an identifying prefix on messages it generates.

Mix

The mix is the set of processes currently in your system, including active, scheduled, and suspended processes, as well as frozen libraries.

These processes include jobs and tasks initiated by users (and you), and jobs and tasks initiated by programs and the operating system.

These topics are covered in this subsection:

- Displaying active entries
- Displaying completed entries
- Displaying scheduled entries
- Displaying the mix
- Setting the mix limit

Displaying Active Entries (A)

Use the A (Active Mix Entries) system command to display the active mix entries on your system.

This command displays only the jobs and tasks that are currently executing; it does not include waiting or scheduled entries or frozen libraries. This is the same display usually included in the automatic display mode (ADM) displays, although you can view pages of active entries with the A command, instead of the 4 or 5 lines in the ADM display.

For more information about the A command, and for an explanation of the command display, see the *System Commands Reference Manual.* For a discussion of ADM mode and of the ADM displays, see "Monitoring Processes at the ODT" in Section 3.

Procedures

- To display all the unsuppressed active entries, enter

  ```
  A
  ```

- To display all the unsuppressed and suppressed active entries, enter

  ```
  A ALL
  ```

- To display only the active entries originating from a specific queue, enter

  ```
  A QUEUE <queue number>
  ```

- To display only the active entries running with a specific usercode, enter

  ```
  A USER <usercode>
  ```

- To cause the active entries display to be sorted in order of decreasing usage of central processor unit (CPU) time, append any *A* command with the following:

  ```
  SORT CPU
  ```

  So, for example, to display all the suppressed and unsuppressed active entries sorted by CPU time, enter

  ```
  A ALL SORT CPU
  ```

  You can include the sort option in an automatic display mode (ADM) option as well:

  ```
  ADM(A ALL SORT CPU 15, MSG) DELAY 9
  ```

Displaying Completed Entries (C)

The system creates a message every time a job or task finishes, and puts these messages in the completed entries list. You can display these entries with the C (Completed Mix Entries) system command.

These entries can be valuable because they contain the reason a job or task finished. This is important if a job or task did not complete normally; these entries can tell you why (this information is also contained in the system log and the job summary report).

The completed entries are the same as those usually included in the automatic display mode (ADM) displays. The C command, however, enables you to see a full page of the entries, instead of just 5 or 6 lines. For a description of ADM and the ADM displays, see "Monitoring Processes at the ODT" in Section 3.

The reasons for termination are listed in the Hist (history) column in the completed entries display. The reasons for termination follow:

Reason	Meaning
?-DS	Unknown discontinue (DS) (history not set)
A-DS	Attribute discontinue (task and job attribute conflict)
D-DS	Data comm discontinued
E-DS	Discontinued by DMS II software
EOJ	Normal end of job
EOT	Normal end of task
F-DS	Fault discontinued (for example, "divide by zero" or "segmented array" error)
I-DS	Input/Output error discontinued
O-DS	Operator-Discontinued
P-DS	Programmatically discontinued
Q-DS	Queue-Discontinued (job or task not compatible with the queue specified)
R-DS	Resource-Discontinued (for example, "print limit exceeded" or "stack overflow")
S-DS	System-Discontinued
SNTX	Syntax error
U-DS	Unknown discontinuation
Unn-DS	(Where nn is a number), Discontinued with a HISTORYCAUSE task attribute value of "nn"

Procedure

To display the completed mix entries, enter

```
C
```

This command displays a full screen of completed entries, with the most recently completed jobs and tasks listed first. If there is more than one screen, use the NS (Next Screen) system command to view following screens, or use the CQ (Clear Queue) system command to discard the rest of the output queued for display.

Example

Suppose a user calls you, wondering what happened to his program. He says the name of the program is TEST/PAYROLL/UPDATE, and his usercode is Walt. He says he ran it 20 minutes ago, but nothing happened. Since the job was initiated 20 minutes ago, you cannot find it in any of the ADM displays (it has been replaced by more recent job entries).

You look at the active entries with the A (Active Mix Entries) system command, but the job is not there. There are no waiting entries, so you use the C command to look at the older completed entries. You enter

```
C
```

You see the following display:

```
-Job-Task-Time--Hist------ COMPLETED ENTRIES --------------------------
1886\1928 12:22 SNTX (PRIYU) *ALGOL (PRIYU)CANDE/CODE59Ø ON FIRE
1925\1925 12:22 EOJ  (NORM) JOB *SYSTEM/XREFANALYZER
1923\1923 12:21 EOJ  (NORM) JOB NEDCOMPILE
1923\1924 12:21 EOT  (NORM) *DCALGOL OBJECT/SYMBOL/NED
1922\1922 12:2Ø O-DS (WALT) JOB TEST/PAYROLL/UPDATE
1855\1918 12:19 EOT  (ROBY) (ROBY)WFLCODE ON PACK
1855\1919 12:19 EOT  (FARZIN) *LIBRARY/MAINTENANCE
1855\1915 12:17 EOT  (ROBY) (ROBY)WFLCODE ON PACK
```

Each completed entry contains, from left to right:

- The job and task numbers
- The time the job or task finished, displayed in 24-hour format
- The reason the job or task finished
- The usercode of the job or task originator (in parentheses)
- The name of the job or task

You note from the display that the program TEST/PAYROLL/UPDATE was operator-discontinued at 12:20. This means that someone used the DS (Discontinue) system command on the program for some reason.

Displaying Scheduled Entries (S)

Use the S (Scheduled Mix Entries) system command to display any scheduled entries (those whose initiation has been delayed) on your system. This command does not display jobs in job queues.

The system schedules processes if it does not have enough memory to fulfill all the job and task initiation requests that have been entered, if the system is searching for disk space to assign to the system log or overlay files, if there is an HS (Hold Schedule) system command in effect, or if there is a pending CM (Change MCP).

If you notice scheduled entries or see any jobs or tasks in the waiting entries with "SUSPENDED BY SYSTEM" or "SECTORS REQUIRED" messages, notify your system administrator.

The HS command is discussed in "Jobs and Tasks" in this section.

Procedures

- To display any scheduled entries that have not been suppressed, enter

```
S
```

- To display all scheduled entries, including suppressed ones, enter

```
S ALL
```

Examples

The following is an example of a scheduled entries display:

```
--Mix-Pri-Elapsed------ 3 SCHEDULED ENTRIES ----------------------------
 1808  50   09:07 (NORM) JOB UTIL/TEST
 1823  50   10:11 (ROBY) OBJECT/SYMBOL/PATCH2
 1829  50   14:22 (RGB) *DCALGOL (RGB)OBJECT/PATCH/RGB
```

Each scheduled entry contains the following fields (from left to right):

- The mix number
- The job or task priority
- The amount of time elapsed since the job or task was scheduled
- The usercode of the job or task originator
- The job or task name

If there is a HS (Hold Schedule) system command in effect, the letters *HS* are included in the first line of the scheduled entries display like this:

```
--Mix-Pri-Elapsed------ 3 SCHEDULED ENTRIES (HS)------------------------
 18Ø8  5Ø   Ø9:Ø7 (NORM) JOB UTIL/TEST
 1823  5Ø   1Ø:11 (ROBY) OBJECT/SYMBOL/PATCH2
 1829  5Ø   14:22 (RGB) *DCALGOL (RGB)OBJECT/PATCH/RGB
```

Displaying the Mix (MX)

Use the MX (Mix Entries) system command to display the jobs and tasks in the mix. This command also shows the most recent DISPLAY and RSVP messages each job or task might have.

Procedures

- To use the command to see all the unsuppressed jobs and tasks, enter

  ```
  MX
  ```

- To see all the jobs and tasks, including suppressed ones, enter

  ```
  MX ALL
  ```

- You can display only the jobs and tasks originating from a specific job queue by entering

  ```
  MX QUEUE <queue number>
  ```

  The queue number is the number of the queue whose jobs and tasks you want to see.

Example

The following is an example MX display:

```
---Mix-Pri---------------- JOB ENTRIES ---------------------------------
  7275 5Ø   (NORM) JOB NEDCOMPILE
  7276 5Ø   ..(NORM) *DCALGOL (NORM)OBJECT/SYMBOL/NED
  7259 8Ø   JOB *SYSTEM/CANDE ON PACK
  7277 5Ø W ..*SYSTEM/DUMPANALYZER
       R:NO FILE MEMORY/DUMP (MT) #1
  7262 8Ø   ..*CANDE/STACKØ1
       D:PK65 (GEORGE)CANDE/RECV6ØØ REMOVED ON FIRE
  7248 8Ø   ..*SYSTEM/COMS ON PACK
       D:DISPLAY: OUTPUT ROUTER SELECTED TO MIX NUM 724
  7253 5Ø   ....JOB MARC
       D:7253 UNISYS A3:615 MARC (VERSION 36.17Ø)
  725Ø 5Ø   ....MARC
       D:725Ø Ø9:48 FROM MACS AT CHRIS:  1Ø:15 H/L CANCELED
```

The Mix field shows the job or task number. The Pri field shows the job or task priority. The word JOB precedes job entries. The entry names are indented to reflect job structure (dependent tasks are indented).

If a job or task is waiting on something, it has a W in front of the job name (like task 7277 in the example). The reason is listed after the *R:*. The most recent display for each job or task is listed under the job or task name, following the *D:*.

Setting the Mix Limit (ML)

Use the ML (Mix Limit) system command to display, set, or remove the system mix limit. The mix limit is the maximum number of jobs and associated tasks that can be introduced into the mix at one time. At times when the count of jobs and tasks equals or exceeds the mix limit, the system will not initiate any jobs in the job queues.

The system mix limit is independent of the queue mix limits. The system mix limit need not equal the total of all the queue mix limits, and changing or removing the system mix limit does not change any queue limits.

You can have a mix limit of zero (you cannot run any jobs), or no mix limit (you can run any number of jobs), or anything in between. For a discussion of the considerations involved in setting the mix limit, see the *System Administration Guide*.

The following procedures describe how to display, set, and remove the mix limit.

Procedure for Displaying the Mix Limit

The mix limit display includes limit information for every job queue as well as for the system. For every job queue, you see the queue number, the active count (the sum of all waiting, scheduled, and active jobs and associated tasks), the queue mix limit, and the number of jobs queued. The letter D at the left of the display indicates the default queue, if any. The total active jobs and the system mix limit are shown at the bottom of the display.

To see this display, enter

```
ML
```

You see something similar to the following:

```
 QUEUE      ACTIVE      LIMIT     QUEUED
D    Ø           2         1Ø          Ø
     3           Ø          2          Ø
     5           4          4          1
 TOTAL           6       NONE
```

In the above example, the NONE entry under the LIMIT heading represents the system mix limit. There is no mix limit to the number of jobs that can be run.

Procedure for Setting the Mix Limit

To set the system mix limit, enter

```
ML = <number>
```

The number is the mix limit you want.

Procedure for Removing the Mix Limit

To remove the mix limit (to make the mix limit equal to none), enter

```
ML-
```

Note: *If the MCP option AUTORECOVERY (OP 8) is reset, then after any halt/load the system automatically sets the mix limit to 0.*

Next Screen, Displaying (NS)

The NS (Next Screen) system command displays the next screen of information if another system command generates more than a one-screen display.

When a system command display does need several screens, the NS command is included in the display at either the upper left-hand corner or the lower right-hand corner. All you have to do is transmit it.

When you enter a command that generates more than one screen of output, that output is queued at the ODT. You must clear this queued output either by displaying it with the NS command, by entering another system command, or by discarding it with the CQ (Clear Queue) system command. If you do not clear this output, the ODT is blocked from displaying ADM and system messages.

The CQ command is described in this section under "Queued Output, Clearing from the ODT (CQ)."

Procedure

To display the next screen of information generated by another command, enter

```
NS
```

If there is no more information, the following message is displayed:

```
NO NEXT SCREEN
```

ODT Mode, Transferring to (??ODT)

The ??ODT (Return from Menu-Assisted Resource Control) system command returns the system control terminal to ODT mode from MARC mode or from a message control system (MCS). This command works only if the SCT is presently in MARC mode or if it is currently owned by an MCS.

When the SCT is in MARC mode, it is controlled by a program called COMS/ODT/DRIVER. All commands entered from the SCT are interpreted by this program and MARC. When you enter the ??ODT command, the SCT is given back to CONTROLLER and the operating system, and commands entered from the SCT in ODT mode are processed directly by CONTROLLER and the operating system.

Procedure

To return to ODT mode from MARC mode, do the following:

1. Log off of MARC. From the Choice line in the HOME menu, enter

   ```
   BYE
   ```

2. On the top line of the MARC log-on screen, enter

   ```
   ??ODT
   ```

The ODT should now be back in ODT mode.

Print System (PrintS)

The Print System (PrintS) handles the printed output generated by the jobs and tasks running on your system. The tasks in this section describe the procedures you need to follow in dealing with printed output. For further information about the Print System, refer to the *Print System Guide.*

The A Series Print System (PrintS/ReprintS) is made up of two software products:

- PrintS, the programs included in your A Series system software that handle printing at onsite printers
- ReprintS, an optional product that enables you to print on remote (data comm) devices and on devices controlled by other host systems in a BNA network

Printed output is an essential product of any computer system, even those systems where interactive use is extensive. Printed output can include end-user reports and summaries, compiler listings, and memory dumps, to name just a few examples.

Because so much information is required in printed form for so many different purposes, a printing system must be flexible enough to satisfy the needs of many different users. Equally important, it must produce output in a timely fashion without slowing down other work being done on the computer.

Printing (or punching) can take place without adversely affecting other processing because of an operation called *spooling*. Spooling uses a fast peripheral, usually a disk or tape, as an intermediate storage area for data being transferred from main storage to printers or punches, which run more slowly. Once stored on disk or tape, the output data can be printed without interfering with main memory or processor operations.

With spooling, the printing operation is separated from the execution of the program that created the output. This separation has several advantages, as follows:

- A program can run through to completion without having to wait for a printer to become available.
- A program can finish executing sooner because it is not delayed by the slower speed of the printer.
- Printers can run continuously, provided there is output to be processed, whereas a printer that is dedicated to one program is idle while the program performs computations.

You can configure the Print System to spool printer files only, punch files only, or both. By default, the Print System enables spooling of both printer and punch files. To disable spooling of printer files, you must assign the system option LPBDONLY the value FALSE. To disable spooling of card punch files, you must assign the system option CPBDONLY the value FALSE.

If you decide not to use spooling, printer output files are routed directly to the first available output device, not to a backup medium. This process is called *direct output*. Direct output links a printer to a running program for the duration of program execution. It is preferable to spooling if you must have the output immediately.

However, direct output has some serious disadvantages, as follows:

- If no suitable printer is available when the program first requests one, the program must wait until a printer is available.
- The printer is dedicated to the program until the program finishes execution. As a result, the printer is idle while the program performs other processing.
- The program takes longer to finish execution because it is delayed by the slower speed of the printer.
- You cannot use many of the features that customize printed output.

You can specify direct output for individual jobs, tasks, or files even when the system as a whole is set up for spooling. To specify direct output, you use the PRINTDISPOSITION file attribute.

When spooling is used, the printer or punch output file is written to a backup medium, and the file is referred to as a *backup file.* (Printer backup files are also called *BD files*). A backup file is a temporary copy of a program output file. In addition to the data to be printed, printer backup files contain all the carriage control information specified by the program, so that the final output can be properly formatted.

These files, by default, are stored on the halt/load family. Other files, such as the master control program (MCP) code file and the system files, are also stored on the halt/load family by default. So unless your site has changed the location where these files are stored, there might not be much room on the halt/load family. To store print requests on another family, use the DL (Disk Location) system command.

A *print request* consists of one or more files to be printed, along with the attributes that describe how to print those files. The Print System combines all the files with similar characteristics from a job, session, or task into print requests. Backup files are grouped into different print requests if they have incompatible attribute values; for example, different destinations, different forms requirements, or different times to be printed.

The Print System constructs print requests for files that it considers capable of being printed, barring unforeseen circumstances. For example, it constructs print requests for backup files spooled to disk and for files named in a Work Flow Language (WFL) PRINT statement.

After constructing a print request, the Print System places it in a queue for printing. Once a print request has been placed in the print queue, it can be controlled and modified by the PS (Print System) system commands. When all conditions are met (for example, the right type of device is available, any special forms are in the printer, and any specified date or time for printing has arrived), the system starts printing the file or files.

If printing finishes without incident, the backup files are removed from disk unless you have used the SAVEBACKUPFILE file attribute to request that they be retained. If the LOCKEDFILE file attribute is TRUE, whether the backup files are removed is determined by the value assigned to the REMOVELOCKEDFILES option of the Print System.

If, however, an irrecoverable error such as a disk read error prevents a file from being printed completely, the Print System marks the print request as an *exception*. It puts the exception request back in the print queue and does not try again to process it. Exception requests stay in the queue until you intervene to retry or to delete them.

For certain files, the Print System does not construct print requests. The Print System does not keep a record of backup files written to tape. Therefore, these files remain on tape until you use the SYSTEM/BACKUP utility to print them, or until you copy them to disk and use either the PS ADDFILES command or the WFL PRINT statement to print them.

By assigning the value DONTPRINT to a file's PRINTDISPOSITION attribute, you indicate that you want to print the file at some later, unspecified time rather than letting the Print System control the printing. Therefore, the Print System does not create a print request.

If you specify direct output for a file, printing takes place while your program is executing and there is no need to construct and queue a print request. Direct output files are invisible to the Print System.

Configuring the Print System

Configuring the printing environment is usually considered to be an administrative activity, whereas monitoring and controlling of printing devices and print jobs is an operations activity. However, the distinctions between administration and operations activities can vary from site to site. For example, at a small site the same person might perform both administration and operations tasks. At a large site, the system administrator might design the configuration of the printing environment, but delegate the actual implementation of that design to an experienced operator.

Generally, system administrators are responsible for the following tasks:

- Initializing the Print System
- Setting up the spooling process
- Setting the maximum number of print servers
- Specifying where backup files are written
- Establishing criteria for print request selection
- Specifying the following defaults for the Print System:
  - Whether the SYSTEM/BACKUPFILELIST file is compressed at each halt/load
  - Whether the Print System automatically creates and configures unknown remote devices
  - The default destination group for printing requests without a specific destination
  - The criteria used for printing job summary files
  - The default PAGECOMP declaration used to format job summary files
  - The default PAGECOMP declaration used to format backup files
  - The point when a print request is generated for a file
  - The printer type to be used for printing backup files
  - The removal of backup files for which the LOCKEDFILE file attribute has the value TRUE
  - The largest identification number that can be assigned to a print request
  - The ordering of print requests shown by a PS SHOWREQUESTS command
- Configuring output devices by doing the following:
  - Making them known to the Print System
  - Specifying PRINTERKIND values to indicate the supported page description languages
  - Specifying control users or stations
  - Setting volume limits and limit times
  - Configuring printers for special forms
  - Setting the page size (number of lines each form includes)

  - Specifying a default PAGECOMP declaration to be used on a device
  - Specifying whether a printer should take checkpoints, and setting the checkpoint interval
  - Setting up virtual servers to offline output devices
  - Specifying a transform function to be used on a printing device
  - Specifying whether a printer should suppress header and trailer pages by default
- Establishing the default destination printer group and other device groups
- Associating printing devices with specific users, specific input devices, or both
- Specifying the content and format of banner, header, and trailer pages
- Setting PRINTDEFAULTS by usercode
- Creating a load file of Print System commands

The configuration tasks in the preceding list are summarized in the *System Administration Guide* and fully documented in the *Print System Guide.* If your system administrator has delegated any Print System configuration tasks to you, refer to these guides.

If your site is using the optional and separately priced Remote Print System (ReprintS) to control printing at remote destinations, or if you want to use the font capabilities provided by certain printers, your system administrator is responsible for certain other tasks discussed in the *Print System Guide.*

Once the Print System has been configured to meet the needs of your site, you should normally be concerned only with querying the configuration to help you resolve printing problems. For information about this task, see "Displaying the Configuration of the Print System" later in this section.

Creating a Load File of Print System Commands

Because the configuration of printers can be very complex and the Print System has many features that enable you to customize it for your site, you might want to create a file of Print System commands to automate these tasks. This type of file is called a *load* file. The load file named *PRINTS/STARTUP on the DL BACKUP family is automatically processed after a restart of the Print System (or a halt/load), before any print requests are initiated. You can process any load file at any other time by using the PS LOAD command.

Commands are processed according to the privileges of the user entering the PS LOAD command until an error is reported or until the end of the file is reached. Because the *PRINTS/STARTUP file is processed after the FONTSUPPORT library is initialized but before any print requests exist, that load file should not contain any commands that act upon print requests. Only the beginning and ending of processing is reported; intermediate responses for the individual commands are not provided.

A load file must have the value of TEXTDATA for its FILEKIND file attribute. The file can contain any Print System command (including other PS LOAD commands), and each command (including the last one in the file) must be terminated by a semicolon (;). A command can continue across more than one line, but tokens must be contained on a single line. The keyword *PS* is optional for the commands contained in a load file.

Strings can be concatenated with the ampersand (&) concatenation operator. Text following a percent sign (%) outside of a quoted string is treated as a comment. You can add blank lines to improve readability.

For example, you might have a load file named (SYSADMIN)PS/SETUP that has the following contents:

```
% Specify the desired Print System defaults:
PS DEFAULT COMPRESSFILELIST=TRUE;
PS DEFAULT CONFIGURE=STRICT;
PS DEFAULT DESTINATION=LINEPRINTERS;
PS DEFAULT JOBSUMMARY=CONDITIONAL;
PS DEFAULT PRINTDISPOSITION=EOJ;
PS DEFAULT REMOVELOCKEDFILES=AUTOMATIC;
PS SELECTION BY PRINTPRIORITY, VOLUME;

% Initialize the font library:
PS LOAD (SYSADMIN)PS/SETUP/FONTLIB;

% Configure printers, create groups, and associate devices:
PS CONFIGURE LP1 PRINTERKIND=LP, CONTROL=STATION REMOTE/SPO,
                 LIMIT MAX 100000, CHECKPOINT 1000 LINES,
                 TRAILER=SUPPRESSED;
PS CONFIGURE LP2 PRINTERKIND=LP, CONTROL=STATION REMOTE/SPO,
                 LIMIT MIN 5000, CHECKPOINT 1000 LINES,
                 TRAILER=SUPPRESSED;
PS CONFIGURE REMOTE/HPLJ2 PRINTERKIND=PCL4, PCL3, LINEPRINTER,
                          LIMIT MAX 5000, TRAILER=SUPPRESSED,
                          TRANSFORM="PCL4" & " IN SL DEVICESUPPORT",
```

```
                           FONTSET=HPLJ2;
PS GROUP LINEPRINTERS=LP1, LP2;
PS ASSOCIATE REMOTE/HPLJ2 WITH USER VIP;

% Activate print servers and remote printing:
PS SERVERS=3;
PS REMSERVERS +;
PS BNA +;

% Define header and banner pages:
PS HEADER 10 SPACE 1,
          20 USERCODE:BLOCK SAME,
          30 SPACE 1,
          40 JOB NUMBER:BLOCK "@",
          50 SPACE 2,
          60 ORIGIN,
          70 REQUESTNUM,
          80 DEVICE,
          90 TIME:LOWER,
          PAGECOMP="PORT CPL=80 LPP=80";
PS BANNER 10 SPACE 1,
          20 FILENAME,
          30 SPACE 1,
          40 USERCODE,
          50 SPACE 1,
          60 JOB NUMBER;
% End of load file
```

To process this load file, enter the following command:

```
PS LOAD (SYSADMIN)PS/SETUP ON PACK
```

The system returns the following response:

```
Starting batch processing of file (SYSADMIN)PS/SETUP ON PACK
Batch processing successful.  44 records processed.
```

Displaying the Configuration of the Print System

You can use several Print System commands to display different elements of the Print System configuration. The basic displays, the commands that invoke them, and the information contained in these displays are included in Table 8–2.

Table 8–2. Displaying Elements of the Print System Configuration

Display	Command	Information in the Display
Output device associations	PS ASSOCIATE	• User/device name with association • Devices in the association
BNA ReprintS information	PS BNA	Whether it is enabled or not
Default Print System settings	PS DEFAULT	• Compression of the SYSTEM/BACKUPFILELIST file • Automatic configuration of unknown remote devices • Default destination group • Criteria for printing job summary files • PAGECOMP declaration for job summary files • PAGECOMP declaration for backup files • Determining when a print request is generated • Printer type for printing backup files without a PRINTERKIND • Removal of locked backup files • Maximum print request identification number • Ordering of PS SHOWREQUESTS displays
Print Device Characteristics	PS DEVICES	For each device: • Device type (LP, IP, CP, and so on) • Accessibility (default, DESTINATION only, and so on) • Request limits (maximum lines, minimum lines, and so on) Other elements are included if applicable.

continued

Table 8–2. Displaying Elements of the Print System Configuration (cont.)

Display	Command	Information in the Display
Device groups	PS GROUPS	• Names of groups known to system • Names of devices belonging to each group
Header, banner, trailer information	PS HEADER PS BANNER PS TRAILER	Content items, spacing items, and PAGECOMP declaration
Print request selection criteria	PS SELECTION	The criteria (a combination of volume, priority, and request number) that determine print request selection
Print servers	PS SERVERS	• Maximum servers • Servers in use

The preceding table summarizes the information available to you about Print System configuration. Refer to the *Print System Guide* for complete information about Print System configuration displays.

Information about the Print System configuration helps you understand how the Print System is working. For example, if a print request is waiting for a line printer even though all the line printers are available, you might find out why with information about how the line printers are configured. It might be that all the line printers have a maximum line number limit for print requests enabled to print on them, and the waiting print request is too large. Or maybe someone has set the print server limit to zero.

Just knowing what kind of information you have access to can give you ideas about solving a problem or about better tuning the Print System to meet the needs of your site. You can experiment with the display commands to see what is there to help you. Becoming familiar with the *Print System Guide* is also helpful.

Procedures

- To display the input device and user associations defined to the system and the output devices in each association, enter

  ```
  PS ASSOCIATE
  ```

- To display the status of BNA print routing, enter

  ```
  PS BNA
  ```

- To display the default Print System settings, enter

  ```
  PS DEFAULT
  ```

- To display printer device characteristics (such as device type, accessibility, limits), use the following commands:

Command	**Display**
PS DEVICES	All the printing devices known to the system.
PS DEVICES <device name>	A specific print device. The device name is the name of the device you want to see the characteristics for (LP4 is an example of a device name).
PS DEVICES <group name>	A print device group. The group name is the name of the print device group you want to see the characteristics for. The command to see what groups are defined on your system is described in these procedures.

- To display the names of the device groups known to the system along with the devices belonging to each group, enter

```
PS GROUP
```

You can list the devices in a particular group by entering

```
PS GROUP <group name>
```

- To display the current format description for the header, banner, and trailer pages, use these commands:

Command	**Display**
PS HEADER	The header page (the optional cover page indicating the beginning of a print request)
PS BANNER	The banner page (the optional separator page between files within a print request)
PS TRAILER	The trailer page (the optional last page indicating the end of a print request)

- To display what determines the order in which print requests are selected for printing, enter

```
PS SELECTION
```

- To display the number of print servers enabled on the system and the number presently in use, enter

```
PS SERVERS
```

Examples

- You want to see the device characteristics of the line printer LP4, perhaps to find out what, if any, the request limits are. You enter

```
PS DEVICE LP4
```

The system responds with the following display:

```
LP4
     Status:           Idle
     Device Type:      Printer
     Accessibility:    In default Group
     Request limits:   Minimum 5000 lines, Maximum 20000 lines
     Header/Trailer:   Both printed by default
     Printerkind:      LINEPRINTER
     Available Fonts: None, no PDL PRINTERKINDs are configured
```

The display shows you that LP4 is a default printer, that only print requests between 5,000 and 20,000 lines are enabled to print on it, and that it is currently idle.

- A user calls you, and says that she is running a job that generates several printed reports. She says that the job has finished with two of the reports, but she cannot find them anywhere in the system. They have not printed, and she cannot see them in any of the PS displays.

  You decide to find out what the default PRINTDISPOSITION is. You enter

```
PS DEFAULT
```

The resulting display looks like this:

```
Default            Value              Description
------------------------------------------------------------------------
COMPRESSFILELIST   FALSE          BACKUPFILELIST not compressed at H/L.

CONFIGURE          STRICT         Printers must be configured before use.

DESTINATION        LP             Default destination group.

JOBSUMMARY         ABORTONLY      Job Summary is printed if there is
                                  an abnormal termination.

PRINTDISPOSITION   EOJ            Requests generated when job terminates.

REMOVELOCKEDFILES  NEVER          Locked files cannot be removed by Print
                                  System.

PRINTERKIND        DONTCARE       Backup files printed on any printer.

REQUESTMAX         9999           Maximum usable Request Number.

SHOWDISPLAY        SORTED BYTIME  Most recent requests appear first.

PAGECOMP   (PC)    FEED=A PAPERSIZE=LETTER FONT DEFAULT=COURIER(10 CPI)
                   LANDSCAPE PSET=1 STACK=1 LPP=66 COLOR=BLACK CPL=132

JOBSUMMARY PC      FEED=A PAPERSIZE=LETTER FONT DEFAULT=COURIER(10 CPI)
                   LANDSCAPE PSET=1 STACK=1 LPP=66 COLOR=BLACK CPL=132
```

You note that the PRINTDISPOSITION is EOJ (which stands for end of job). You tell the user that the reports will print when the entire job completes, that is, when the job generating the reports reaches end of job.

Displaying the Status of the Print System (PS STATUS)

You can display the status of the Print System with the PS (Print System) STATUS system command. The PS STATUS display includes the following information:

- The number of print requests printing
- The number of print requests waiting
- The number of exception print requests
- The number of remote print requests stopped

This display gives you a quick overview of the activity on the Print System. You can include this display in the automatic display mode (ADM) displays by using the PS option. For an explanation of ADM, see "Monitoring Processes at the ODT" in Section 3.

Procedure

To display the status of the Print System, enter

```
PS STATUS
```

Example

The PS STATUS display looks similar to the following:

```
                          Print System Status
------------------------------------------------------------------------
          3 requests printing.
          1 requests waiting.
          4 requests in EXCEPTION state.
          No remote requests stopped on a printer.
```

Handling Print Requests

A *print request* consists of one or more files to be printed, along with the attributes that describe how to print those files. The Print System combines all the files with similar characteristics from a job, session, or task into print requests. Backup files are grouped into different print requests if they have incompatible attribute values; for example, different destinations, different forms requirements, or different times to be printed.

The Print System constructs print requests for files that it considers capable of being printed, barring unforeseen circumstances. For example, it constructs print requests for backup files spooled to disk and for files named in a Work Flow Language (WFL) PRINT statement.

After constructing a print request, the Print System places it in a queue for printing. Once a print request has been placed in the print queue, it can be controlled and modified by the PS (Print System) system commands. When all conditions are met (for example, the right type of device is available, any special forms are in the printer, and any specified date or time for printing has arrived), the system starts printing the file or files.

If printing finishes without incident, the backup files are removed from disk unless you have used the SAVEBACKUPFILE file attribute to request that they be retained. If the LOCKEDFILE file attribute is TRUE, whether the backup files are removed is determined by the value assigned to the REMOVELOCKEDFILES option of the Print System.

If, however, an irrecoverable error such as a disk read error prevents a file from being printed completely, the Print System marks the print request as an *exception*. These requests show a status of "E:" or "*Exception*" in the PS SHOWREQUESTS display.

The Print System puts the exception request back in the print queue and does not try again to process it. Exception requests stay in the queue until you intervene to retry or to delete them.

The following tasks describe the basic things you can do with print requests, both to modify them and to deal with problems.

Deleting Print Requests (PS DELETE)

You can delete print requests from the print queue, or individual files from a print request, with the PS DELETE command. Files associated with a deleted request are treated as if they had been printed.

Normally, data files are retained and temporary printer backup files are deleted after being printed. However, a printer backup file is not deleted in the following circumstances:

- Its SAVEBACKUPFILE file attribute is assigned the value TRUE
- Its LOCKEDFILE file attribute is assigned the value TRUE
- It resides on a CD-ROM disk
- It resides on a disk that is write-protected

You can select a set of print requests to delete based on any of the following:

- A list of print request numbers, which you can obtain from the PS SHOWREQUESTS displays
- The ALL clause, to select every request currently in the print queue
- The ALL EXCEPTIONS clause, to select every exception request currently in the print queue
- A <selection expression> containing a combination of a BEFORE time and the values of the following task and file attributes:
  - ACCESSCODE
  - DESTINATION
  - FORMID
  - JOBNUMBER
  - MIXNUMBER
  - USERCODE

To delete an individual file within a print request, enter the PS DELETE command followed by a colon (:) and either the file number or the file title. You can obtain this information by including the *:FILES* option at the end of a PS SHOWREQUEST command.

Procedures

- To remove a list of print requests, enter

  ```
  PS DELETE <request list>
  ```

  The request list is a list of print request numbers.

- To remove all print requests, enter

  ```
  PS DELETE ALL
  ```

This command removes all print requests, including exception requests.

- To remove all the exception requests, enter

  ```
  PS DELETE ALL EXCEPTIONS
  ```

  This removes all the exception requests, but no others.

- To remove a set of print requests having a particular value for selected task and file attributes, enter

  ```
  PS DELETE <selection expression>
  ```

  For information about deleting selected print requests, see the *Print System Guide.*

- To delete the second file in print request number 14000, enter

  ```
  PS DEL 14ØØØ:2
  ```

Displaying Print Requests (PS SHOWREQUESTS)

You can display information about the print requests in your system with the PS (Print System) SHOWREQUESTS command. You can display information about specific print requests if you know their request numbers. You can also display the main print request list, which includes all print requests (except those that are scheduled or have recently completed), or you can display a subset of print requests selected according to their status:

- BNA
- Completed
- Exception
- Forced
- Printing
- Scheduled
- Waiting

For any print request list, you can specify a <selection expression> to further restrict the display based on a combination of a BEFORE time and the values of the following task and file attributes:

- ACCESSCODE
- DESTINATION
- FORMID
- JOBNUMBER
- MIXNUMBER
- USERCODE

You can combine these attributes in a single PS SHOW command, but each attribute can have only a single value. The display of recently completed print requests cannot be restricted based on FORMID or on a destination at another BNA host.

Procedures

- To display the main print request list, enter

  ```
  PS SHOW ALL
  ```

- To display the BNA print request list, enter

  ```
  PS SHOW BNA
  ```

- To display the completed print requests, enter

  ```
  PS SHOW COMPLETED
  ```

  This command can be abbreviated as PS SH C.

- To display the exception print requests, enter

  ```
  PS SHOW EXCEPTIONS
  ```

  This command can be abbreviated as PS SH E.

- To display forced print requests that are waiting for a device, enter

  ```
  PS SHOW FORCED
  ```

  This command can be abbreviated as PS SH F.

- To display the currently printing print requests, enter

  ```
  PS SHOW PRINTING
  ```

  This command can be abbreviated as PS SH P.

- To display the scheduled print requests, enter

  ```
  PS SHOW SCHEDULED
  ```

  This command can be abbreviated as PS SH S.

- To display the waiting print requests, enter

  ```
  PS SHOW WAITING
  ```

  This command can be abbreviated as PS SH W.

Examples

- The following is an example of the main print requests list:

```
PS SH ALL

                            Print Request List
Job  Request Pri  Amt Left Done  (User) Request Name          Status
---- ------- ---  -------- ----  ---------------------------  -------
3587   13235  5Ø    15Ø Ln  4Ø%  (MYCODE) MYWRITE             P: on LP2
Ø612   13282  5Ø    435 Ln       (HERCODE) INFO/WRITE/NEW     W: CP29
5595   13613  5Ø   9Ø26 Ln       (HISCODE) "Session"          W: TR LP
67Ø2   13984  65     14 Pg  77%  (UCODE) MONTHLY/REPORTS      P: on LP7
6Ø84   14ØØØ  5Ø    957 Ln       *JOB/TRANSFER                T: A15AB
4263   14Ø26  5Ø    328 Ln       (SITE) JOB/DECK              F: LP7
5824   14Ø39  5Ø   3672 Ln       (OPS) DAILY/LOG              E: A15AB
```

- The following is an example of a BNA Requests display:

```
PS SH B

                     BNA Requests
Job  Request Pri   Amount   (User) Request Name
---- ------- ---  --------  ---------------------------------------
1234   10234  50      54 Ln  (FRED) COMPILE/PRINT/SUPPORT
    Sent to BNA host A15AB (Request # 8972)
    Printing on LP4                      30% complete
1109   10291  50      19 Ln  (SITE) JOB/DECK
    Ready to transfer request to BNA host A15AB
    Waiting for BNA host A15AB to respond.
8372   10293  50     955 Ln  (JANE) OMNI/OUTPUT/BD
    Sent to BNA host A12CD (Request # 5422)
    Ready to print.
6754   23457  50    2534 Ln  (OPS) SITE/REPORT
    Sent to BNA host A15AB (Request # 238)
    *Exception*.
3875   25774  50    7855 Ln  (OPS) DAILY/LOG
    Sent to BNA host A12CD (Request # 3980)
    Transferring files.
```

- The following is an example of a Recently Completed Print Requests display:

```
PS SH C

          Recently Completed Print Requests
Job  Request  (User) Request Name        Time  Printer
---- -------  -------------------------- ----- -------
3770    2445  (PROG1) "Session"          09:05 LP4
0665    2444  (FRED) OBJECT/PRINTLP ON   08:36 PA158
1890    2443  (PAYROLL) OBJECT/SUMMARY O 07:07 LP4
3609    2442  (SITE) STARTUP             07:05 LP4
```

Note that the job name is truncated to fit in the display.

- The following is an example of a Print Request Exceptions display:

```
PS SH E

                   Print Request Exceptions
Job  Request Pri   Amount   (User) Request Name
---- ------- ---  --------  --------------------------------------------
1148    3496  50      62 Ln  (UTIL)OBJECT/UTILITY/DTRS ON OPS
    Error - P602B break on output - powered off or disconnected.
9981    3443  50      62 Ln  (DCMAINT) (UTIL)OBJECT/UTILITY/DTRS ON OPS
    Error - P602B break on output - powered off or disconnected.
9981    3441  50      62 Ln  (DCMAINT) (UTIL)OBJECT/UTILITY/DTRS ON OPS
    Error - P602B break on output - powered off or disconnected.
4737    3268  50       1 Pg  (STAUB) "Session"
    Operator requeued request on PA238.
5010    3152  50      14 Ln  (SITE) "Session"
    Ready to transfer request to BNA host LFA15CD
```

- The following is an example of a Forced Requests display:

```
PS SH F

                          Forced Requests
Job  Request Pri  Amt Left  (User) Request Name            Status
---- ------- ---  --------  ----------------------------- -------------
5513   13604  50    174 Ln  (MYCODE) OBJECT/PRINTS         F: PA271
4233   12231  50   5546 Ln  (NEWCODE) PL/NEW/EXAMPLE       F: LP4
```

- The following is an example of a Currently Printing Requests display:

```
PS SH P

                    Currently Printing Requests
Job  Request Pri  Amt Left Done  (User) Request Name         Device
---- ------- ---  -------- ----  -------------------------   ----------
2587   13144  50     36 Pg  62%  (MYCODE) MYWRITE            LP7
4612   13282  50    557 Ln  33%  (HERCODE) INFO/WRITE/NEW    CP29
5595   13413  50   9026 Ln   0%  (HISCODE) "Session"         LP1315AB
3587   13235  50    150 LN  40%  (MYCODE) BIG/JOB/OUTPUT     AP1
     Cover Open
5595   13613  50    435 Ln  12%  (HISCODE) "Session"         AP2
     Out of Paper
0612   13282  50    711 Ln  10%  (HERCODE) HERWRITE          AP3
     Not Ready
1234   13245  50    333 Ln  97%  (THEIRCODE) "Session"       AP4
     Jammed
```

- The following is an example of a Scheduled Print Requests display:

```
PS SH S

                        Scheduled Print Requests
Job  Request Pri   Amount   (User) Request Name        Sched time
---- ------- ---  --------  -------------------------  -----------------
0640   11975  50    948 Ln  (SITE) COPY                19:00 on 11/15/88
    Scheduled for LP7, a Line Printer
0354   11971  50    828 Ln  (PROG) WEEKLY/DBDUMP       20:00 on 11/15/88
    Scheduled for LP4, a Line Printer
0359   11976  50    527 Ln  (USER1) TEST/FORMS         20:00 on 11/15/88
    Scheduled for STATION LPG158, an Image Printer
```

- The following is an example of a Waiting Print Requests display:

```
PS SH W

                  Waiting Print Requests
Job  Request Pri   Amount   (User) Request Name
---- ------- ---  --------  ------------------------------------------
5513   13604  50    174 Ln  (MYCODE) OBJECT/PRINTS
    Waiting for any Printer
4233   12231  50   5546 Ln  (NEWCODE) PL/NEW/EXAMPLE
    (Forced) Waiting for LP2004AB, a Line Printer
    From BNA Host A12A.
    Ready to print.
```

- The following example shows how you can restrict a print request list using a <selection expression>:

```
PS SHOW  USER=PAYROLL, ACCESS=CONTROL, DEST=LP9, FORMID="PAYCHECKS"

PS SHOW COMPLETED  DESTINATION=MY_OWN_AP, USER NEQ BIGBILL

PS SHOW WAITING  D LP4, BEFORE 10:30

PS SHOW WAITING  MIX 5634
```

Forcing Print Requests to Print (PS FORCE)

You can make a print request print before other requests with the PS (Print System) FORCE command. A forced request is printed as soon as an appropriate print device is available.

The PS FORCE command is useful when you need the output from a job to print immediately, but there are other print requests ahead of your request. This command makes the specified print request the first in line to print. As soon as a printer is available, that request prints.

You must also use this command if a print request was marked as an exception request because a print device failed while that request was printing. If the device is working again, you must use this command to clear the exception and place the request at the front of the queue.

The PS FORCE command does not change task attribute, file attribute, or print modifier specifications. For example, if a request requires an image printer but your system does not have any image printers, you cannot force the request to a line printer. In such a case, you must use the PS MODIFY command to change the value of the PRINTERKIND file attribute.

However, the PS FORCE command does override the AFTER file attribute. Also, a forced request is not constrained by any size limit and time limit specifications for a device. You can force a print request even if the number of print servers allowed on the system is zero (that is, if PS SERVERS = 0).

You can specify either a print request number or a job number in the PS FORCE command, but not both. If you specify a job number, every print request generated by that job is forced, even if the command is entered before the job finishes. Job numbers and request numbers are obtained from the PS SHOWREQUESTS displays.

Procedures

- To force a print request to print as soon as possible, enter

  ```
  PS FORCE <request number>
  ```

  The request number is the number of the print request you want to print. Request numbers can be displayed with the PS SHOWREQUESTS command, as described in “Displaying Print Requests (PS SHOWREQUESTS)” in this subsection.

- To force a job to print as soon as possible, enter

  ```
  PS FORCE JOB <job number>
  ```

  The job number is the number of the job you want to print. Job numbers can be displayed with the PS SHOWREQUESTS command, as described in “Displaying Print Requests (PS SHOWREQUESTS)” in this subsection.

Handling Exception Print Requests

Exception print requests are print requests that have been marked with an exception flag. An exception flag is an event that makes it impossible for a request to print.

Use the PS SHOWREQUESTS EXCEPTIONS command to display the exception print requests on your system. This command is described in "Displaying Print Requests (PS SHOWREQUESTS)" in this subsection.

Exception requests will not print or leave the Print System without intervention. You can handle most exceptions with the PS MODIFY, the PS FORCE, or the PS DELETE command. These commands are described in this subsection in "Modifying Print Requests (PS MODIFY)," "Forcing Print Requests to Print (PS FORCE)," and "Deleting Print Requests (PS DELETE)," respectively.

For detailed information about handling exception print requests, see the *Print System Guide*.

Modifying Print Requests (PS MODIFY)

The PS (Print System) MODIFY system command enables you to change the file attributes and print modifiers of one or more print requests. A print request can require modification for a variety of reasons, but the most common are because it has been marked as an exception, or is not being accepted for printing. For example, if a print request is assigned to a printer in the default destination group, but the printer is down for maintenance, you can use this command to direct the request to another printer.

Print requests that are not being accepted for printing usually have an incorrectly specified file attribute. A print request can also be marked as an exception because of problems with a backup file or a printing device, a PS REQUEUE or a PS SKIP command, or incompatible software levels.

You can modify the value of any file attribute except LOCKEDFILE or PRINTPARTIAL, and any print modifier except REQUESTNAME, either by specifying a different value or by restoring the system default value. (To change the value of the LOCKEDFILE file attribute, you must use the WFL ALTER statement.) For information on file attributes, see the *File Attributes Reference Manual*. For information on print modifiers, see the *Print System Guide*.

You can change any number of file attribute or print modifier values with one PS MODIFY command. You can select a set of print requests to modify based on any of the following:

- A single print request number or list of print request numbers, which you can obtain from the PS SHOWREQUESTS displays
- The ALL clause, to select every request currently in the print queue (including scheduled requests)
- The ALL EXCEPTIONS clause, to select every exception request currently in the print queue
- A <selection expression> containing a combination of a BEFORE time and the values of the following task and file attributes:
  - ACCESSCODE
  - DESTINATION
  - FORMID
  - JOBNUMBER
  - MIXNUMBER
  - USERCODE

Procedures

- To retry print requests without any modification, use the appropriate one of the following commands.
  - For specific print request numbers:

    ```
    PS MODIFY <request list>
    ```

  - For all print requests:

    ```
    PS MODIFY ALL
    ```

  - For all exception requests:

    ```
    PS MODIFY ALL EXCEPTIONS
    ```

    You can abbreviate EXCEPTIONS as E.
  - For a set of print requests selected by the value of certain task and file attributes:

    ```
    PS MODIFY <selection expression>
    ```

- To restore the system default value for a file attribute or print modifier, precede the name of the attribute or modifier by a minus sign (–).
  - For specific print request numbers:

    ```
    PS MODIFY <request list> - <print modifier or file attribute>
    ```

  - For all print requests:

    ```
    PS MODIFY ALL - <print modifier or file attribute>
    ```

  - For all exception requests:

    ```
    PS MODIFY ALL EXCEPTIONS - <print modifier or file attribute>
    ```

    You can abbreviate EXCEPTIONS as E.
  - For a set of print requests selected by the value of certain task and file attributes:

    ```
    PS MODIFY <selection expression> - <print modifier or file attribute>
    ```

- To change the value for a file attribute or print modifier, assign the desired value using the equal sign (=).
  - For specific print request numbers:

    ```
    PS MODIFY <request list> <print modifier or file attribute> = <value>
    ```

- For all print requests:

  ```
  PS MODIFY ALL <print modifier or file attribute> = <value>
  ```

- For all exception requests:

  ```
  PS MODIFY ALL EXCEPTIONS <print modifier or file attribute> = <value>
  ```

  You can abbreviate EXCEPTIONS as E.

- For a set of print requests selected by the value of certain task and file attributes:

  ```
  PS MODIFY <selection expression> <print modifier or file attribute>
  = <value>
  ```

Example

Assume that you have one default printer, LP4, and that the printer has failed for some reason. Assume also that you have another printer, LP7, which is usually used to print short printouts.

Since LP4 is the default printer, all print requests with no value for the DESTINATION attribute are assigned to this printer. Since this printer is down, all these requests are waiting to print. The request printing on LP4 when it went down became an exception print request.

You enter a PS SHOW WAITING command to look at the waiting print requests. You see the following display:

```
PS SH W

                  Waiting Print Requests
Job  Request Pri   Amount   (User) Request Name
---- ------- ---  --------  ------------------------------------------
7826    2607  50      47 Ln  (SITE) *OBJECT/CHECKLIST ON USERDISK
    Waiting for LP4, a Line Printer
5254    2609  50     546 Ln  (USER1) *OBJECT/TESTFILE ON USERDISK
    Waiting for LP4, a Line Printer
6827    2611  50     380 Ln  (USER1) JOB/SUMMARY ON USERDISK
    Waiting for LP4, a Line Printer
6853    2620  50     114 Ln  (SITE) *OBJECT/BACKUPLIST ON DISK
    Waiting for LP4, a Line Printer
```

The first, second, and fourth print requests (requests 2607, 2609, and 2620) are needed, and cannot wait for LP4 to come back up. You want to reroute these requests to LP7 so that they can print. To do so, you change the DESTINATION attribute for these requests with the following command:

```
PS MODIFY 2607, 2609, 2620 DESTINATION = "LP7"
```

There is still the request that was printing on LP4 when it failed. You enter a PS SHOW EXCEPTIONS command to look at this request:

```
PS SH E

               Print Request Exceptions
Job  Request   Amount   (User) Request Name
---- -------  --------  ----------------------------------------------
Ø261    4989   8Ø46 Ln  (USER1) "Session"
    Error - device LP4 failed during printing.
```

This printout is not needed soon, so you simply retry the request to include it in the requests waiting for LP4. To do so, you enter

```
PS MODIFY 4989
```

This command clears the exception status of the request, and retries it. Since LP4 is not working yet, it simply waits for that printer. If this printout is needed immediately, you can modify the destination as you did for the waiting requests.

Ordering Print Request Displays

You can choose to have the list of print requests that is displayed in response to a PS SHOW command ordered in two ways: unsorted (the default), or ordered in accordance with the time each request was queued in the Print System. If you choose the ordered display, the print request most recently queued in the print system is displayed first, the second most recent print request is displayed second, and so on.

You can specify a display method for the entire Print System through the SHOWDISPLAY option of the PS DEFAULT command, or you can specify a display method for an individual PS SHOW command by including the BYTIME option in the individual command. The SHOWDISPLAY option affects the display order for all the PS SHOW commands, except the PS SHOW COMPLETED command. The SHOWDISPLAY value is stored over halt/loads.

Procedures

- To display the current setting of the SHOWDISPLAY option, enter the following command:

  ```
  PS DEFAULT SHOWDISPLAY
  ```

- To instruct the entire Print System to sort the print requests output in PS SHOW displays according to the time the print requests entered the print queue, enter the following command:

  ```
  PS DEFAULT SHOWDISPLAY BYTIME
  ```

- To return to the default PS SHOW displays of unsorted print requests, enter the following command:

  ```
  PS DEFAULT SHOWDISPLAY UNSORTED
  ```

- To sort the output of a single PS SHOW display in accordance with the time the print requests entered the queue, use the following command modification:

  ```
  <any PS SHOW command> BYTIME
  ```

- To cause a single PS SHOW command display to be unsorted, use the following command modification:

  ```
  <any PS SHOW command> UNSORTED
  ```

Redirecting a Print Request

Programmers can specify a destination for a program's output within the program, using file and task attributes. Operators, however, do not usually deal with the output of jobs and tasks until after the jobs and tasks create print requests.

Note: *Each print request has a request number. The request number is not the same as the mix number of the process that created the request.*

Normally, you redirect an *exception print request*, which is a print request that the system was unable to print for some reason. You can, however, redirect any print request before it begins printing.

Procedure

To direct a print request to a particular print device, use the following syntax:

```
PS MODIFY <request number> TO DESTINATION = "<device name>"
```

The request number is the number of the print request you want to direct. You can get this number from the main print request list, or the exception print request list, displayed by the PS SHOW and PS SHOW EXCEPTION commands, respectively.

The device name is the name of the device you want the print request sent to. You can get a list of device names for a particular device type by entering *PER LP* or *PER IP*, as described in "Displaying Device Status (PER)" earlier in this section.

Example

If you want to direct print request 4589 to the print device LP7, you enter

```
PS MODIFY 4589 DESTINATION = "LP7"
```

Requeuing Print Requests (PS REQUEUE)

The PS (Print System) REQUEUE system command stops a printing print request, prints a separator page, marks the print request as an exception, and returns the print request to the waiting queue. This command is valuable when, for example, the paper is misaligned in a printer or when there is a problem with a printer ribbon. The PS REQUEUE command enables you to stop the print request, fix the printer, and reprint the request.

You can stop a currently printing request by specifying either its request number or the name of the printing device. After the currently printing request has been requeued, the device is ready to service another print request.

You can restart printing of the request by using either the PS FORCE or PS MODIFY system command. The PS FORCE and PS MODIFY commands are described in this subsection under "Forcing Print Requests to Print (PS FORCE)" and "Modifying Print Requests (PS MODIFY)," respectively.

You can specify a print request to be requeued immediately, at the next page break, or at the next end-of-file (EOF) control code. No matter where a print request is requeued, a separator page is printed.

The point at which a requeued request begins printing when it is reselected depends on whether you specify the optional CHECKPOINT clause. If you specify the CHECKPOINT clause (or if the printer has been configured to take mandatory checkpoints), printing begins from the point in the file where the request was requeued. If you do not specify the CHECKPOINT clause, the file is printed from the beginning.

When you requeue a print request that has been routed to a remote host, you identify the print request by the request number that is assigned by the local host. The command is passed to the remote host, which determines the corresponding request number that is assigned to that print request. The command is then executed on the remote host, but no response is returned to the local host.

Procedures

- To requeue a print request immediately (that is, at the end of the current line), enter one of the following commands:

```
PS REQUEUE <device name>
PS REQUEUE <request number>
```

- To begin printing at the point where printing stopped, specify the CHECKPOINT clause in the PS REQUEUE command by using one of the following commands:

```
PS REQUEUE <device name> CHECK
PS REQUEUE <request number> CHECK
```

- To requeue a print request at the next page break, specify the AT PAGE clause by using one of the following commands:

```
PS REQUEUE <device name> AT PAGE
PS REQUEUE <request number> AT PAGE
```

- To requeue a print request at the end of the file currently being printed, specify the AT EOF clause by using one of the following commands:

```
PS REQUEUE <device name> AT EOF
PS REQUEUE <request number> AT EOF
```

Scheduling Print Requests

You can schedule a print request to print at a specific time and date with the AFTER file attribute. This is useful when you have a large printout that you do not need right away; you can schedule the print request to print after hours so that it does not use printer resources when others might need them.

There are two ways to specify a print time with the AFTER file attribute:

- When you run a program, include a file equation with the AFTER attribute for the program's output file in the RUN statement.
- Use the PS MODIFY command to assign a value to the AFTER attribute of a print request after the request has been created.

When you file equate a program's output file with the AFTER attribute, the program runs normally, but when the print request is created, it is scheduled to print at the time specified with the attribute. To use file equation, however, you need to know the internal name of the program's output file.

To use the PS MODIFY command, you need to modify the print request when it is waiting to print. It would be too late if the request is already printing. You can modify a print request that is already scheduled, to either change the scheduled time or to cancel the scheduled time.

When the scheduled time for a print request arrives, the request is placed in the waiting queue to print when an appropriate print device becomes available.

Procedures

- To schedule a printout in the RUN statement of a program, use the following syntax:

  ```
  RUN <program name>;FILE <output file name>(AFTER = "<print time>")
  ```

  The program name is the name of the program you want to run and whose output you want to schedule. The output file name is the internal name of the program's output file. The print time is the time you want the output to print. The print time has the format hh:mm, using the 24-hour clock (for example, AFTER = "17:30"). You can also include a date, using the format ON mm/dd/yy (for example, AFTER = "17:30 ON 4/15/92"). The date is optional.

- To schedule a print time for an existing print request, or to change the print time for a scheduled print request, enter the following:

  ```
  PS MODIFY <request number> AFTER = "<print time>"
  ```

  or

  ```
  PS MOD <request number> - AFTER
  ```

You can display request numbers with the PS SHOWREQUESTS command, which is described in "Displaying Print Requests (PS SHOWREQUESTS)" in this subsection. The print time is the time you want the print request to print. The print time has the format hh:mm, using the 24-hour clock (for example, AFTER = "04:30"). You can also include a date, using the format ON mm/mm/yy (for example, AFTER = "04:30 ON 4/15/92"). The date is optional.

- To remove the scheduled time for a print request, enter:

```
PS MODIFY <request number> AFTER = ""
```

The <request number> is the number of the request for which you want to remove the scheduled time. You can display the request number of scheduled requests with the PS SHOWREQUESTS SCHEDULED command, which is described in "Displaying Print Requests (PS SHOWREQUESTS)" in this subsection.

Examples

- If you want to schedule the output from the program PRINT/TEST, using a file equation in the RUN statement, you enter

```
RUN PRINT/TEST;FILE OUT(AFTER = "18:ØØ")
```

In this example, the internal name of PRINT/TEST's output file is OUT. The output is scheduled to print at 6:00 p.m.

- If you want to schedule a print time for an existing print request (for the example, request number 5817), you enter

```
PS MODIFY 5817 AFTER = "Ø6:ØØ ON 5/7/92"
```

This example schedules request 5817 to print at 6:00 a.m., May 7, 1992.

- If you want to cancel the scheduled time for request number 5817 so that it prints as soon as possible, you enter

```
PS MODIFY 5817 AFTER = ""
```

Printing Backup Files Unknown to the Print System

At any particular time, the Print System only knows about files that have information in the SYSTEM/BACKUPFILELIST file. These are backup files, generated when a task is run, that require printing.

The Print System does not know about a backup file if one of the following conditions is true:

- The file has been spooled to tape.
- The file has been spooled to disk with the PRINTDISPOSITION file attribute set to DONTPRINT.
- The file has been spooled to disk and printed, but retained where it was created because the SAVEBACKUPFILE file attribute is TRUE.
- The file has been copied to disk using library maintenance from tape or another system.

To print backup files that are no longer recorded in the SYSTEM/BACKUPFILELIST file, you must use one of the following:

- SYSTEM/BACKUP utility
- WFL PRINT statement
- PS ADDFILES command

Using the SYSTEM/BACKUP Utility

Using the SYSTEM/BACKUP utility is the easiest way to print backup files that reside on tape. With this utility, you can route files directly to a printer or use the spooling process. Also, you can print every file on a tape, all files on a tape starting from a specified file, or an individual file. Refer to the *Printing Utilities Guide* for complete information about the use of this utility.

Using the WFL PRINT Statement

You can use the Work Flow Language (WFL) PRINT statement to print files that reside on disk, but are unknown to the Print System. The PRINT statement enables you to print individual files and file directories. In addition, you can specify file attribute and print modifier values.

Using the PS ADDFILES Command

You can use the PS ADDFILES command to print all files in the *BD (backup disk) and *BP (backup punch) directories that are unknown to the Print System. The PS ADDFILES command causes the Print System to search every disk family on the system and create print requests for all files not recorded in the SYSTEM/BACKUPFILELIST file. Backup files unknown to the Print System are then added to the Print System tables and an end-of-job (EOJ) control code is generated for the job number in each added BD or BP file title.

If a job number is still active (the session is still logged on or a job is still running with that job number), the PS ADDFILES command does not print those backup files. Those backup files are printed with the rest of the session output. To print a single file, or even just a few, use the WFL PRINT statement.

Files from the *BD and *BP directories are grouped into print requests according to job number and file attribute values. File attribute values are obtained from the backup file's control record. Backup files must have standard system-generated titles to be printed with the PS ADDFILES command.

The syntax for the PS ADDFILES command is as follows:

```
PS ADDFILES ALL
```

You can abbreviate ADDFILES as ADD.

Readying Print Devices

The PS (Print System) READY command makes a print device that has been saved with a PS (Print System) SAVE command available again for printing.

Note: *A site printer that was saved with an SV (SAVE) system command must be readied with an RY (READY) system command instead of the PS READY command.*

Procedure

To ready a print device, use the following syntax:

```
PS READY <device name>
```

The device name is the name of the print device you want to ready.

Example

Suppose a line printer, LP4, has been saved so that no new requests can print on it while a new ribbon is put in. The printer is now ready to print, so you enter

```
PS READY LP4
```

The printer is then available for printing.

Saving Print Devices

The PS (Print System) SAVE command prevents the Print System from considering a print device as a destination. This command has no effect on a print request if the request is already printing, but prevents other requests from starting printing.

Print requests specifically routed to a saved printer wait in the printer queue until the printer is readied and again available for printing.

This command is useful when you wish to make a print device unavailable to the system, possibly for maintenance.

Procedure

To save a printer, and make it unavailable for printing new requests, use the following syntax:

```
PS SAVE <device name>
```

The device name is the name of the device you want to make ineligible for printing. No new print requests can print on the device until it is readied with a PS READY command.

Example

Suppose you want to prevent any new requests from printing on printer LP5, but you want the currently printing request to finish, because you want to put a different sized paper in the printer. You enter the following command:

```
PS SAVE LP5
```

The request already printing finishes, and then no new requests will print until the printer is readied with the PS READY command.

Starting and Stopping a Printing Device

You can stop a printing device with the PS (Print System) STOP system command. This command can either stop a print device immediately or stop it at the end of the file that is currently printing. This command is helpful when you need to make minor adjustments to a printer.

Use the PS (Print System) OK system command to restart the printing device.

To restart a print request that has been routed to a remote host, identify the print request by the request number that is assigned by the local host. The command is passed to the remote host, which determines the corresponding request number that it assigned to that print request. The command is then executed on the remote host, but no response to the command is returned to the local host.

Procedures

- To stop a printing device immediately, enter either of the following commands:

  ```
  PS STOP <device name>
  PS STOP <request number>
  ```

  The device name is the name of the printer you want to stop. The request number is that of the print request you want to stop.

- To stop a printing device at the end of the file currently printing, enter either of the following commands:

  ```
  PS STOP <device name> EOF
  PS STOP <request number> EOF
  ```

  The device name is the name of the printer you want to stop. The request number is that of the print request you want to stop.

- To restart a print device that was stopped with a PS STOP command, enter either of the following commands:

  ```
  PS OK <device name>
  PS OK <request number>
  ```

  The device name is the name of the printer you want to restart. The request number is that of the print request you want to stop.

Example

Suppose the ribbon on line printer LP3 is beginning to fade. The printer is in the middle of a long printout, request number 1234. You want to stop the printer and replace the ribbon. You enter one of the following commands:

```
PS STOP LP3
PS STOP 1234
```

The printer stops at the end of the line it was printing. You replace the ribbon, and then enter the following command to restart the printer:

```
PS OK LP3
```

The printer resumes printing where it left off.

Starting and Stopping the Print System

The PS (Print System) QUIT and the PS (Print System) RESTART system commands enable you to change versions of the Print System or PRINTSUPPORT library without disrupting the rest of the system.

After you enter a PS QUIT command, no new requests for use of the Print System, its utilities, or the PRINTSUPPORT library are accepted. All print servers terminate immediately, and if you are running MARC, the command causes MARC to delink from the PRINTSUPPORT library.

While the Print System is stopped, programs requiring access to the PRINTSUPPORT library (mainly those creating printer backup files) display the message

```
WAITING FOR PRINTSUPPORT TO INITIALIZE
```

These programs resume as soon as the Print System is started again.

You should keep all Print System files on the same disk family. If you use the SL command to assign PRINTSUPPORT to a different disk family, you must move all Print System files to that newly assigned disk family to be able to print.

The disk files SYSTEM/BACKUPFILELIST and SYSTEM/PRINTERINFO should not be changed or removed when the Print System is stopped. If these files are changed or removed, the Print System might not be able to recover when it is restarted. (The SYSTEM/BACKUPFILELIST file contains a list of all the printer backup files presently on the system, and the SYSTEM/PRINTERINFO file contains all the configuration information for the print system.)

If you are falling back to a different mark release level, specify the target release level to be used, and the Print System reformats the data structures you need. You specify the mark release you are reverting to by using the following command:

```
PS QUIT <mark release number>
```

Procedures

- To stop the Print System, enter

  ```
  PS QUIT
  ```

  All printing stops immediately. Wait for the SYSTEM/PRINT/SUPPORT job to finish before entering the restart command.

- To restart the Print System after it has been stopped with a PS QUIT command, enter

  ```
  PS RESTART
  ```

  Printing resumes where it left off, and programs waiting on the PRINTSUPPORT library also resume. All the print requests in the Print System when it was stopped are processed normally, as are all requests created while the Print System was stopped.

Programs, Running (RUN)

As an operator, you might need to run programs. Use the WFL *RUN* statement to do this. Note that the RUN statement can be used only to execute *object code files*. An object code file is a file produced by a compiler when a program is compiled successfully. If you try to use the RUN statement at the ODT on a file that is not an object code file, you get a "DSED MISSING CODE FILE" message.

You cannot use the RUN statement for WFL jobs (programs written in Work Flow Language), because WFL jobs are not object code files. To execute WFL jobs you must use the *START* statement. The START statement is discussed under "Jobs and Tasks" in this Part.

To run a program you must know the program file name, and the disk the program file is stored on.

The following procedures include instructions on how to use a basic RUN statement, how to include a family substitution statement in a RUN statement, and how to use common file equations in a RUN statement. For a brief explanation of family substitution and of file equation, see "Specifying Alternate File Attributes" in Section 3.

Using the Basic RUN Statement

The RUN statement executes a program. You can track the program in the system messages and the active entries to monitor the progress of the program and to ensure that it is running properly.

If you try to execute a nonexistent file, for example, if you spell the name of the program incorrectly, the job resulting from the RUN statement is placed in the waiting entries with a "NO FILE" message. Discontinue the waiting job if this happens. For information about discontinuing jobs, see "Jobs and Tasks" in this section.

If you try to use the RUN statement on a file that is not an object code file (if you try to use RUN to execute a WFL job, for example), you see the message "NON-EXECUTABLE CODE FILE" in the system messages.

Procedure

To run a program, use the following basic syntax:

```
RUN <file name> ON <family name>
```

The file name is the name of the object code file you want to execute. The family name is the name of the disk family the file is stored on.

Example

To run a program named OBJECT/PAYROLL on a family named BOOKS, enter

```
RUN OBJECT/PAYROLL ON BOOKS
```

The program then enters the active entries:

```
---Mix-Sub-Pri--CPU Time-------- 23 ACTIVE ENTRIES ---------------------
  6755   2  5Ø       :ØØ *OBJECT/PAYROLL ON BOOKS
                          .
                          .
                          .
```

Using Family Substitution in a RUN Statement

You can use a family substitution statement with a run statement. This family substitution directs the operating system to look for disk files used by the program on the specified family or families. For an explanation of the family substitution statement, see "Understanding File Attributes" in Section 3.

Procedures

- To include a substitution family statement in a RUN statement, use the following syntax:

```
RUN <file name> ON <family name>;FAMILY DISK = <substitute family> ONLY
```

  The file name is the name of the program you want to run. The family name is the name of the family where the program is stored. The substitute family is the family name where the files needed by the program are stored. For example, to execute a program named TEST/PROG stored on the family TESTDISK, and to specify that the files needed by TEST/PROG that reference the family DISK are actually on the family named DATA, enter

```
RUN TEST/PROG ON TESTDISK;FAMILY DISK = DATA ONLY
```

- To include a substitution family and alternate family statement in a RUN statement, use the following syntax:

```
RUN <file name> ON <family name>;FAMILY DISK = <substitute family>
OTHERWISE <alternate family>
```

  The file name is the name of the program you want to run. The family name is the family the program is stored on. The substitute family and alternate family are the names of the families where the disk files needed by the program are stored. For example, to execute a program named TEST/PROG stored on the family TESTDISK, and to specify that the files needed by TEST/PROG that reference the family DISK are actually on the family named DATA or on the family named OPERATIONS, enter

```
RUN TEST/PROG ON TESTDISK;FAMILY DISK = DATA OTHERWISE OPERATIONS
```

Using File Equation in a RUN Statement

File equation enables you to specify the values of file attributes when you run a program. For a basic description of the function of file attributes, see "Understanding File Attributes" in Section 3. For complete information about file equation and file attributes, see the *I/O Subsystem Programming Guide.*

Note that you can have more than one file equation in a RUN statement by separating file equations with semicolons.

The following procedures demonstrate file equations for specifying

- Alternate file names
- Specific family names
- Alternate file titles
- Alternate device kinds

Procedures

- To specify an alternate file name for a program file in a RUN statement, use the following syntax:

```
RUN <file name> ON <family name>;FILE <internal file name>
(FILENAME=<external file name>)
```

  The file name is the name of the program you want to run. The family name is the family on which the program is stored, and the internal file name is the name of the logical file declared in the program. The external file name is the name of the file on the family that the program needs. For example, to execute the program TEST/PROG on the disk OPSDISK, and to equate the logical file DATA with the physical data file NEWDATA, enter

```
RUN TEST/PROG ON OPSDISK;FILE DATA(FILENAME = NEWDATA)
```

  The preceding command would cause the program to use the disk file NEWDATA.

- To specify a specific disk family for a logical file in a RUN statement, use the following syntax:

```
RUN <file name> ON <family name>;FILE <internal file name>
(FAMILYNAME = <new family name>)
```

  The file name is the name of the program you want to run. The family name is the family on which the program is stored, and the internal file name is the name of the logical file declared in the program. The new family name is the family on which the physical file is located. For example, to execute the program TEST/PROG, located on the disk family OPSDISK, and to specify that the disk file DATA is located on the family INVENTORY, enter

```
RUN TEST/PROG ON OPSDISK;FILE DATA(FAMILYNAME = INVENTORY)
```

- To specify an alternate file title for a program file, use the following syntax:

  ```
  RUN <file name> ON <family name>;FILE <internal file name>
  (TITLE = <external file name> ON <family name 2>)
  ```

  The file name is the name of the program you want to run. The family name is the family on which the program is stored. The internal file name is the name of the logical file declared in the program. The external file name is the name of the physical file associated with the logical file declared in the program, and the family name 2 is the disk family on which the physical file is stored. For example, to execute the program TEST/PROG, and to specify that the physical file associated with the logical file OUT is named NEWDATA and is located on the family INVENTORY, you would enter

  ```
  RUN TEST/PROG ON OPSDISK;FILE OUT(TITLE = NEWDATA ON INVENTORY)
  ```

- To specify an alternate device kind for a logical file, use the following syntax:

  ```
  RUN <file name> ON <family name>;FILE <internal file name>
  (KIND = <device type>)
  ```

  The file name is the name of the program you want to run. The family name is the family on which the program is stored. The internal file name is the name of the logical file associated with the program. The device type is the type of device for the external file associated with the logical file. For example, to execute the program PROG/TEST, and to specify that the file associated with the program's logical file INPUT is a tape, enter

  ```
  RUN PROG/TEST ON OPSDISK;FILE INPUT(KIND=TAPE)
  ```

Example

Suppose that you need to run a program named PAYROLL/UPDATE, located on the family PAYROLL, which requires an input data file named PAYROLL/DATA. This input file is usually found on the family PAYROLL, but this week the file has been stored on tape, and is named PAY/DATA. Since the program is written to expect the file to be named PAYROLL/DATA and to find the file on PAYROLL, you need to use file equation for the program's input file to tell the program the correct place to find this week's data. The program's input file is named INPUT.

You enter the following file equation statements when you run PAYROLL/UPDATE:

```
RUN PAYROLL UPDATE ON PAYROLL;FILE INPUT(FILENAME = PAY/DATA,
KIND = TAPE)
```

Queued Output, Clearing from the ODT (CQ)

When a command entered at the ODT generates more than one screen of information, the first screen is displayed, and the rest of the information is queued to the ODT. This queued information must be cleared before the ODT can continue normal operation.

One way to clear the queued information is to view it with the NS (Next Screen) system command. The other way is to use the CQ (Clear Queue) system command to discard the waiting screens of information. A description of the CQ command follows. The NS command is described in "Next Screen, Displaying (NS)."

Procedure

To clear information queued to the ODT, enter

```
CQ
```

The system responds with the message

```
MESSAGES FLUSHED
```

Sending Commands to an MCS or Database (SM)

Use the SM (Send to MCS or Database) system command to send control commands to a message control system (MCS) or database.

Specify the MCS or database with its mix number. You can find the mix numbers of the MCSs running on your system by displaying the active entries. You can find the mix number of a database with the DBS (Database Stack Entries) system command.

For example, if you want to find the mix number for CANDE, you look for JOB SYSTEM/CANDE in the active mix entries.

To find the mix number for a database, enter DBS at the ODT, and look under the entry for that database.

The responses from the MCS or database you send a command to show up in the system messages.

Procedure

To send a control command, enter

```
<mix number> SM <user text>
```

The mix number is the mix number of the MCS or database you want to send a message to. The user text is the message you send.

Example

Suppose you want to send the message “HELLO” to all the stations owned by the MCS CANDE. After finding the mix number for CANDE (for the example, 9428), enter:

```
9428 SM SS ALL HELLO
```

SS ALL is the CANDE command to send a message to all stations. The command and command text are passed to CANDE. You can find the response sent back by CANDE in the system messages (either with the MSG command or under the messages entries if your ODT is using automatic display mode):

```
------Mix----Time----------Message-------------------------------------
    *9428    3:12 DISPLAY: #.
```

A pound sign (#) is the CANDE prompt, meaning the command has been accepted.

This command is useful when readying devices owned by an MCS.

Software, Installing

Every time you get a new software release, or buy new software, you need to install it on your system.

To help you with this process, Unisys publishes the *Software Release Installation Guide*, which you should receive with your software release.

The *Software Release Installation Guide* contains product overviews, procedural steps for installing software, and reference data, and you should use the guide whenever you install software.

Since the *Software Release Installation Guide* contains detailed procedural information for the installation process, software installation is not covered in this book.

System

The operator tasks described under the heading "System" deal with the system as a whole, rather than one aspect of it.

The word *system*, when used alone, refers to all the hardware (including processors, memory, and peripherals) and system software (including the MCP, compilers, and support libraries).

The following topics are covered in this subsection:

- Displaying the system configuration
- Displaying system utilization
- Initializing the system
- Powering the system on and off
- Displaying the status of programs using the system

Automatically Displaying the System Status (ADM)

Automatic display mode (ADM) displays the current status of programs using the system, and automatically updates these displays. You can choose what is displayed and how often the display is updated through the ADM command. For an explanation of the ADM (Automatic Display Mode) system command, see "Monitoring Processes at the ODT" in Section 3. For detailed information about each of the ADM settings, see the *System Commands Reference Manual.*

The most basic forms of the ADM command follow.

Procedures

- If you want to start automatic display mode using the default settings, enter

  ```
  ADM
  ```

  The command entered by itself like this starts ADM with the following default settings:

  ```
  ADM (A7, W3, S2, C5, MSG) DELAY 9
  ```

  An explanation of these settings is in Section 3 under "Monitoring Processes at the ODT."

- To stop automatic display mode and retain the present setting, enter

  ```
  ADM ST
  ```

- To resume automatic display mode stopped by an ADM ST command, enter

  ```
  ADM OK
  ```

- To cancel automatic display mode and clear the present setting, enter

  ```
  ADM-
  ```

Displaying System Configuration (GC, SC)

Use the GC (Group Configuration) and SC (System Configuration) system commands to display information about the system's current configuration.

The most common use for this information is finding out why a peripheral is not visible to the system.

The GC display shows you, among other things, what data link processors (DLPs) serve what groups of peripherals. So, if the system cannot communicate with a peripheral, and you cannot find out why, you can use this display to see what DLP it is associated with. You can then check the DLP to see if that is the problem.

The GC display also tells you what peripheral unit numbers are defined. If you get new peripherals, your Unisys engineers need to look at this list so you do not pick a unit number that is not already defined on the system. New unit numbers can be added, but you must add them to the peripheral-configuration description for the system.

The SC display contains the microcode version (if any) installed on your system and the status of memory. If your system is running very slowly, but the number of jobs in the mix is not high, you can look at the memory status in this display to see if the system has access to all of its memory.

Procedures

- To see the Group Configuration display, enter

  ```
  GC
  ```

- To see the current System Configuration display, enter

  ```
  SC
  ```

Examples

This is an example of a GC display:

```
GROUP:  MPA15C

2 PROCESSORS: 4-5    1 IOP: Ø    1 MEMORY UNIT: Ø

CACHESIZE    6 MEGA WORDS
%            (23Ø3 TRACK BUFFERS)

PERIPHERALS:
11Ø2,13ØØ,24ØØ,15ØØ,12ØØ,2ØØØ,11ØØ,11Ø3,1ØØ3,11Ø1
1-3,8,9
28-31 VIA 1ØØ7,13Ø7
63-67,7Ø-75 VIA 1ØØ4,11Ø4
1ØØ,1Ø1,1Ø7,1Ø9-111,115,117-12Ø,159,16Ø,236
24Ø-245,247-251 VIA 13Ø4,13Ø5
******************************
226,21Ø4,11Ø6,44-51,54,55,161,221

DISK LOCATIONS:
  SORT     ON DISK
  CATALOG  ON DISK
  JOBS     ON DISK
  USERDATA ON DISK
  BACKUP   ON PACK
  LOG      ON DISKB
  DPFILES  NO FAMILY SPECIFIED
  IPFILES  ON DISK
  OVERLAY  ON DISKB

HOSTNAME:  MPA15C
```

In the preceding example, the line of asterisks appearing in the Peripheral section of the display indicates the new group information that has been added to the partition after it was reconfigured.

The following is an example SC display:

```
2 DATA PROCESSORS 4-5
1 I/O PROCESSOR   Ø
MSM Ø
     16M WORDS IN USE
MEMORY USAGE:
  SYSTEM          1ØM WORDS
  DISK CACHE       6M WORDS
```

Displaying System Utilization (U)

Use the U (Utilization) system command to display the current system utilization statistics.

These statistics can give you an estimate of the load the system is under at the moment. This estimate can tell you why response time might be slow, whether the system can handle starting a big job, or whether you need to cut down on the number of jobs in the mix because the system is struggling.

The numbers in the U command display represent the percentage of the total system resources used by each component over the last 10 seconds. These numbers can show you if the system resources are working efficiently, or if the system is spending time trying to make room for too many jobs and tasks.

As an operator, you need to pay most attention to the following components of the U display:

TRUE IDLE
: This component represents the percentage of time the processor spent idle. High values mean your system has resources to spare. Low values mean your system is overloaded. If you have a true idle value of 10, for example, it would not be a good idea to start up a big job until the value increases.

FALSE IDLE
: False idle time is time the processor spent idle while waiting for overlay data to transfer from memory to disk or disk to memory. It is called "false" idle because the processor could have been active if the required data areas were in memory when needed. This tends to increase the time it takes for jobs and tasks to process.

: If the false idle value is high, though, it usually means the system is struggling, and response time will be slow. If you notice a high value for this component frequently, you should notify your system administrator.

OTHER PBIT
: This value represents the amount of time the system spent reading data into memory and writing data out of memory due to insufficient memory space; that is, the amount of time spent doing overlay operations. When a job begins processing, the MCP reads the program and its related data into memory, and begins processing it, if there is enough memory for all the jobs in the mix.

: If there is not enough memory, however, when the MCP brings in a new job, it has to make room for it by writing the data for another job to disk, then reading the data for this job in. Then, when it is time to process the first job some more, it has to read the data for it in again, and so on. This type of data moving is related to the OTHER PBIT value.

: So if this value is high, your system is spending a lot of time and resources moving data around instead of processing jobs. This makes everyone's response time slow. If this value is high frequently, contact your system administrator.

USER
: This component tells you what percentage of processor time was spent processing user-submitted jobs and tasks. It is handy to compare the value of USER and the value of the MCP component. This gives you an idea of how much time the system is spending on jobs and tasks as compared to system overhead.

You should look at the utilization display at regular intervals to monitor overall system performance. This display can warn you that the system is overloaded and can give you clues as to why.

Example

The following is an example U display:

```
-------------------------------- SYSTEM UTILIZATION --------------------

---------- CPM Statistics -----------    ------------ IO Statistics ----
User       = 77 %  Initial Pbit =  7 %    User     =    36 IO/Sec (  385 K
IO Finish =  2 %  Other Pbit   =  Ø %    MCP      =    13 IO/Sec (  195 K
MCP        =  2 %  True Idle    = 12 %    Datacom =     4 IO/Sec (    Ø K
Search     =  Ø %  False Idle   =  Ø %    Total    =    53 IO/Sec (  58Ø K
                                                        53 IO - Interrupts

------------ Disk Cache -------------

Total    =    152 Rq/Sec (  27Ø Kb/Sec)
```

Initializing the System

Since initialization procedures can be very specific to the system, and since complete documentation on the subject exists, procedures for system initialization are not in this book.

If you want to install new software, or a new software release, refer to the *Software Release Installation Guide.*

If you want information on cold and cool starts, refer to the *Operating System Installation Guide.*

If your system is not operating properly, try a programmatic halt/load (using the ??PHL command, described in “Halt/Load, Performing (??PHL)” in this section). If this does not work, you need to use the maintenance subsystem to reinitialize the system. Information about the maintenance subsystem is in the installation guide for your machine.

Powering the System On and Off (POWER)

If you have an A 1, A 2, A 3, A 4, A 5, or A 6 system, you can set up an automatic power schedule with the POWER (Power Up/Down) system command that turns your system on and off at prearranged times.

For example, you can have your system turn itself on every morning at 8:00, off every evening at 6:00, and stay off on weekends. You can implement any schedule you wish.

You can also use the command to turn off the system at any time, with or without a delay, and to cancel pending power-off commands.

Following are procedures for displaying, setting, and clearing the power schedule, as well as for using the POWER command to directly turn off your system, and for canceling pending power-off operations.

Procedure for Displaying the Power Schedule

To display the automatic power schedule (if any), enter:

```
POWER SCHEDULE
```

A power schedule display looks something like this:

```
         SUNDAY  MONDAY  TUESDAY  WEDNESDAY  THURSDAY  FRIDAY  SATURDAY

ON         -      7:ØØAM  7:ØØAM   7:ØØAM     7:ØØAM    7:ØØAM  8:3ØAM
OFF        -     1Ø:ØØPM 1Ø:ØØPM  1Ø:ØØPM    1Ø:ØØPM    6:ØØPM  6:ØØPM
DELAYOFF  2Ø      2Ø      2Ø       2Ø         2Ø        2Ø       45
```

You also see warning messages if any of the following are true:

- The ON and OFF times of any day are the same.
- Any day's OFF time precedes the ON time.
- Any day's ON time is within an hour of the OFF time for that day.
- Any day's OFF time plus its power-off delay time cross a day boundary and overlap the next day's ON time.

None of these situations causes an error, but each one can cause unexpected power-on or power-off operations, or prevent expected ones.

Some disk drives cannot be powered down remotely. Issuing a POWER OFF command to an M9710 (SCSI/674) disk pack results in the message "PKxx [<serial number>] <packname> CANNOT BE REMOTELY POWERED DOWN."

Procedure for Setting the Power Schedule

You can set power-on power-off times for a day or group of days. You can set the ON and OFF times with the same command or separately:

- To set an ON time for one day, use this syntax:

```
POWER SCHEDULE <day of week> ON AT <time>
```

- To set an OFF time for a single day:

```
POWER SCHEDULE <day of week> OFF AT <time>
```

- To set ON and OFF times for a day with a single command:

```
POWER SCHEDULE <day of week> ON AT <time> OFF AT <time>
```

- To set ON or OFF times for a series of days:

```
POWER SCHEDULE <day of week> THRU <day of week> ON AT <time>

POWER SCHEDULE <day of week> THRU <day of week> OFF AT <time>
```

- To set ON and OFF times for a series of days with the same command:

```
POWER SCHEDULE <day of week> THRU <day of week> ON AT <time>
OFF AT <time>
```

You might have noticed the DELAYOFF line in the example display. When the system is powered off with the power schedule, the system has a delay time between the scheduled power-off time and when the system actually loses power. The delay allows time for warning interactive users and for jobs waiting in queues to finish. The system does not wait for a zero mix; it shuts off at the end of the delay time.

The system uses the delay time to send messages indicating that the system is going to lose power. Messages are sent at the following intervals:

- If the delay time is greater than 10 minutes, a message is sent every 10 minutes.
- If the delay time is less than 10 minutes, but greater than 5 minutes, two messages are sent, with 5 minutes between messages.
- If the delay time is less than 5 minutes, two messages are sent; one at the beginning of the delay time, the other 1 minute before power loss.

The default delay time is 20 minutes. You can set the delay time with the POWER SCHEDULE DELAYOFF command. A delay time of zero is not allowed.

The delay time command syntax is similar to the syntax to set ON and OFF times.

- To set delay times for a single day, enter

```
POWER SCHEDULE <day of week> DELAYOFF <delay time>
```

- To set delay times for a series of days, enter

```
POWER SCHEDULE <day of week> THRU <day of week> DELAYOFF <delay time>
```

The delay time cannot be zero.

Setting or clearing delay times will not change the power schedule on/off times.

For example, if you want to set the delay time for Sunday to 30 minutes, and the delay times for the rest of the week to 15 minutes, you enter

```
POWER SCHEDULE SUNDAY DELAYOFF 30

POWER SCHEDULE MONDAY THRU SATURDAY DELAYOFF 15
```

You can confirm the new delay times with the POWER SCHEDULE command.

Procedure for Clearing the Power Schedule

You can clear the automatic power schedule of any ON or OFF times that are set, and reset all the power-off delay times to the default by entering

```
POWER SCHEDULE CLEAR
```

You can clear the power schedule for one day, or a series of days as well:

```
POWER SCHEDULE <day of week> CLEAR

POWER SCHEDULE <day of week> THRU <day of week> CLEAR
```

These commands do not clear the power schedule for days you do not include in the command.

Procedure for Turning Off the System with a POWER Command

You can turn off the system either immediately or with a delay with a POWER command.

Note: *This command does not wait for jobs or tasks to finish. Make sure the system is in an idle state, or user jobs will be interrupted.*

- To immediately turn off the system, enter

  ```
  POWER OFF SYSTEM
  ```

  This command turns off the system right away.

- To turn off the system after a delay, enter

  ```
  POWER OFF SYSTEM DELAYOFF <delay time>
  ```

The delay time is the number of minutes before the system shuts off. Warning messages are sent just as with a scheduled power-off operation. If you do not include a delay time in the command, the default time of 20 minutes is assumed.

Procedure for Canceling a Pending Power-Off Operation

You can cancel a scheduled or unscheduled power-off operation with this command:

```
POWER OFF SYSTEM CANCEL
```

The command stops a pending power-off operation. It does not affect the automatic power schedule, other than to override a power-off operation one time.

This command works only if there is a pending power-off operation (if the system is in the power-off delay time cycle and issuing warning messages).

Tapes

The following tasks describe common operations involving tapes. Tasks describing operations on files, such as copying files to and from tape, are described under "Files" in this section.

Magnetic tape volumes are of two basic types: labeled, and unlabeled. A labeled tape has a volume label and header labels at the beginning of the tape, which the system reads for information about the contents of the tape. An unlabeled tape has no tape identification information; the tape simply begins with data for the first file. Note that the SN (Serial Number) system command, described below, creates a labeled tape.

Labeled tapes are easier to deal with and to keep track of than unlabeled tapes. You can load a labeled tape on a tape drive and find out the tape's name, serial number, and the files recorded on it.

Programs can automatically read and write from labeled tapes as soon as the tapes are loaded, because the system matches the requested name and serial number with the name and serial number on the tape volume. If the names and numbers match and the tape device is ready, reading or writing begins immediately.

When you copy files to a tape with library maintenance, you must always use a labeled tape. Library maintenance creates a tape directory on the tape. Then, when you read files from that tape, the system uses the directory to access the files. Because the system creates the directory when the copy is initiated, you cannot add files to the end of a tape with another COPY command. See "Copying Files (COPY, ADD)" in this section for more information on copying.

The following topics are covered in this subsection:

- Assigning serial numbers to tapes
- Designating final tape reels
- Displaying tape directories
- Purging tapes
- Unloading tapes

Assigning Serial Numbers to Tapes (SN)

Use the SN (Serial Number) system command to assign serial numbers to tapes.

If you have a group-coded record (GCR) tape subsystem, you can use the SN command to specify recording density. A GCR tape drive can be used in phase-encoded (PE) mode, with a recording density of 1600 bits per inch (bpi), or in GCR mode at 6250 bpi. You cannot specify recording density with other tape subsystems.

One way the MCP identifies a tape mounted on a tape drive is through the tape's serial number. The serial number is simply a unique number stored in the tape's label. Using serial numbers helps prevent accidental uses of the wrong tape and prevents a tape from being appropriated by the wrong job or task.

You should use serial numbers on your system. They provide much better security and control when dealing with tapes. You can tell the master control program (MCP) to require serial numbers by default with MCP option 27 (SERIALNUMBER). When this option is set, whenever you write to a tape you must either specify a tape's serial number or use the OU (Output Unit) system command to open the tape so that you can write on it. For information on setting MCP option 27, see "Setting and Resetting MCP Options (OP)" in this section. For a description of the OU command, see "Directing Output to a Device (OU)" in this section.

> **Caution**
>
> The SN command purges tapes. Make sure you do not need any information on a tape before you use the SN command on it.

Some computer departments assign each new tape a permanent serial number, which remains with the tape throughout its usage. The tape is given a specific serial number according to the numbering system set up by the system supervisor. The serial number appears both in the tape name and on the external tape reel to minimize confusion. The serial number stays associated with that particular tape for the length of its use.

Note: *If your computer department uses tapes with permanent serial numbers, do not use the SN (Serial Number) system command to purge a tape and change the tape serial number. Instead, use the PG (Purge) system command. The PG system command erases the tape contents, but does not change the tape name or erase the tape serial number.*

For more information on the Purge command, see "Purging Tapes (PG)" in this section.

Procedures

Put a write ring on the tape, mount it on a tape drive, and make sure the drive is ready. You can use the PER MT command to check the tape's present serial number (if any), and its name. The PER command is described in "Displaying Device Status (PER)" in this section.

- To purge a tape and assign it a serial number, enter:

  ```
  SN MT <unit number> <serial number>
  ```

  The unit number is the unit number of the tape drive the tape is mounted on. The serial number is the serial number you want to assign to the tape.

  A serial number consists of up to six letters or numbers, or a combination of both. The date is often used as a serial number in the form of *mmddyy* (for example, 10/6/92 would be 100692).

- To purge a tape, assign it a serial number, and specify a recording density (for GCR tape subsystems), enter

  ```
  SN MT <unit number> <serial number> (<density>)
  ```

  This command is identical to the previous command, with the density specification added. The density can be either 1600 for PE mode recording or 6250 for GCR recording.

Example

Suppose you have a tape named OLDDATA with some files you no longer need, and you want to assign it a new serial number and purge it, in preparation of copying some new files to it.

You mount the tape on tape drive 28 and enter a PER MT command to make sure you have the right tape. You see the following PER MT display:

```
------------------------------ MT STATUS -----------------------------
     28*P     [100590] 1600 #1 1:0 <10/05/90> OLDDATA
     29*P     [100290] 1600 #1 1:0 <10/02/90> TESTPROG1
```

The tape you mounted on drive 28 is the correct tape. You also note the asterisk (*) beside the unit number, confirming that the tape device is write-enabled. (If you forget to write-enable the tape device, you see the message "WRITE LOCK OUT" when you enter the SN command.) Enter

```
SN MT28 100692
```

The system responds with the following message:

```
MT28 WILL BE SN-ED
```

When the system is finished purging the tape and assigning it the new serial number, it puts the following message in the message entries:

```
MT28 PURGED
```

When you assign a serial number to a tape that contains files protected with the LOCKEDFILE file attribute, the SN operation does not take place immediately. An operator RSVP message appears in the waiting entries, giving you time to verify the tape contents. Since you know this tape contains old data, you enter

```
<mix no> OK
```

The SN operation then purges the tape and assigns it the new serial number.

You verify the serial number by entering another PER MT command:

```
------------------------------ MT STATUS -----------------------------
     28*P     [100692]  S C R A T C H
     29*P     [100292] 1600 #1 1:0 <10/02/90> TESTPROG1
```

The tape is now a labeled tape; it has been purged and has a new serial number. Programs or library maintenance can now use it.

Designating Final Tape Reels (FR)

The FR (Final Reel) system command specifies that the tape reel just read by a program from a tape unit is the final reel of an unlabeled tape file. You can also use this command as a reply to certain library maintenance messages such as "RECOPY REQUIRED" to indicate that no more files should be copied from or to a particular unit.

When a program is reading an unlabeled tape file, the system cannot determine how many reels the file takes up. So when the system reaches the end of an unlabeled tape volume, the system asks for the next reel.

When the system has read the last reel, you must use the FR command to tell the system that it has read the entire file.

The system asks for the next reel by placing the job or task that is reading the file into the waiting entries with a "NO FILE" message. When the last reel is read, you answer the waiting entry with the FR command, as described in the following procedure and example.

Procedure

To declare the input reel just read by a job or task as the final reel, enter

```
<mix number> FR
```

The mix number is the number of the job or task waiting with the "NO FILE" message.

Example

Suppose you are running a job called OBJECT/GETDATA that is reading an unlabeled data file named TESTDATA from a tape on drive 28, and the file takes up two reels. You have already put the second reel on after the second "NO FILE" message (the first "NO FILE" message is for the first reel), and it has finished. Now, another "NO FILE" message is in the waiting entries for your job, but there are no more reels for the file.

The waiting entries message looks like this:

```
---Job-Task-Sub-Pri---Elapsed------ 1 WAITING ENTRY --------------------
   767/ 767   2  5Ø       :26 JOB (SITE)OBJECT/GETDATA
       NO FILE TESTDATA (UNLABELED MT) #3
```

The message "NO FILE TESTDATA (UNLABELED MT) #3" means the job is looking for the third reel of the file. You tell it the reel it just read is the final one with the FR command:

```
767 FR
```

The job then becomes active and continues processing.

Displaying Tape Directories (TDIR)

Use the TDIR (Tape Directory) system command to list the tape directory (a list of the disk files that have been copied to that tape) of a library maintenance tape mounted on a specific tape drive or having a specific tape name. You can direct the directory listing to the ODT or to a printer.

The TDIR command invokes the utility SYSTEM/FILEDATA to create the directory listing.

Procedures

- To get the tape directory at the ODT, enter

  ```
  TDIR SPO <unit number>
  ```

  The unit number is the unit number of the tape drive the tape is mounted on.

  Or you can use the tape name by entering

  ```
  TDIR SPO <tape name>
  ```

  The tape name is the name of the tape you want the directory for. You can get the tape name with the PER MT command.

- To send the tape directory to the printer, enter

  ```
  TDIR <unit number>
  ```

  The unit number is the number of the drive the tape is mounted on.

  Or you can use the tape name by entering

  ```
  TDIR <tape name>
  ```

  The tape name is the name of the tape you want the directory from. In a PER MT display, the name of a tape volume is the first identifier in the file name displayed as

  ```
  <tapename>/<filename>
  ```

  (For library maintenance tapes the file name always takes the form FILEnnn, where nnn is usually 000.) You can get the name of the tape with the PER MT command.

Example

To list the tape directory of the tape volume named TEST1, mounted on drive 28, you can enter either of the following:

```
TDIR SPO 28

TDIR SPO TEST1
```

The resulting output looks like this:

```
A SERIES ØØ4 REPORT OF 1Ø/7/92 AT 15:48:36.
VERSION 3.Ø.ØØØ
TAPE = TEST1/FILEØØØ. ON UNIT   28
SERIAL # = Ø4Ø592 CREATED 4/Ø5/92
9-TRACK (PE) 16ØØ BPI

(USER)TESTFILE/1,
(USER)TESTFILE/2,
(USER)TESTFILE/3,


               TAPEDIRECTORY INPUT WAS:


      "TDIR SPO 28"
```

There are three disk files on the tape under the usercode USER: TESTFILE/1, TESTFILE/2, and TESTFILE/3.

Purging Tapes (PG)

Use the PG (Purge) system command to purge (erase) the contents of a specified tape.

This command does not change the tape name or serial number. You should use this command instead of the SN (Serial Number) system command if your operation assigns permanent serial numbers to tapes. Note that you can purge a tape and change the tape's serial number at the same time with the SN system command, as described in "Assigning Serial Numbers to Tapes (SN)" in this subsection.

A tape must have a write ring on it before you can purge it. It also must have a serial number, even though the serial number will not be changed.

If you have a group-coded record (GCR) tape subsystem, you can specify tape recording density with this command as well.

You can protect disk and tape files from being accidentally overwritten, removed, or changed by setting the LOCKEDFILE file attribute for each file to TRUE. If you try to purge a protected tape, an operator RSVP message appears on the ODT screen, asking you for confirmation of the PURGE command. You can then either accept the tape purge or cancel the job.

You can find more information about the LOCKEDFILE file attribute in the *File Attributes Reference manual.*

Procedure

To purge an unprotected tape, do the following:

1. Load the tape on a tape device.
2. Write-enable the tape device.
3. Verify that you have the right tape by entering

   ```
   PER MT
   ```

   Look at the entry for the unit number of the drive your tape is mounted on, and make sure you have the correct tape.
4. Enter the PG command:

   ```
   PG MT <unit number>
   ```

   The unit number is the number of the drive your tape is mounted on.

The tape should now be purged and ready to use for copying files.

Some tape files are protected with the LOCKEDFILE file attribute, and need an extra confirmation step before they can be purged. If your tape was not purged, look in the waiting job entries. Your purge job might be waiting for you to confirm it.

5. To accept the purge, you enter

```
<mix number> OK
```

To reject the purge, cancel the job with the DS command.

The tape is now purged and ready to use.

Examples

- Suppose you want to purge the files from the tape OLD/STUFF. You put a write ring on the tape, and mount it on drive 28.

  You enter a PER MT command to make sure this is the right tape. The display looks like this:

  ```
  ------------------------------- MT STATUS ------------------------------
       28*P    [100690] 1600 #1 1:0 <10/02/90> OLD/STUFF
  ```

  It is the right tape, so you enter the PG command:

  ```
  PG MT 28
  ```

  The system responds with the following message:

  ```
  MT28 WILL BE PURGED
  ```

  And then, in the system messages, you see

  ```
  MT28 PURGED
  ```

- Suppose you want to purge protected files from the tape PROTECTED/STUFF. You put a write ring on the tape and mount it on drive 26.

  You enter a PER MT command to make sure this is the right tape. The display looks like this:

  ```
  ------------------------------- MT STATUS -----------------------------
       26*P    [050692] 1600 #1 1:0 <10/02/90> PROTECTED/STUFF
  ```

  This is the right tape, so you enter the PG command. An operator RSVP message appears in the waiting job entries. You respond to the RSVP with the Y command:

  ```
  7068 Y
  ```

  A message similar to the following appears:

  ```
  MT26 CONFIRM PURGE PROTECTED/STUFF [050692]
  ```

  You want to purge the tape, so you enter

  ```
  7068 OK
  ```

  The tape should now be purged and ready to use for copying files.

Rewinding and Unloading Tapes (RW)

The RW (Rewind) system command rewinds and unloads a tape unit.

It is a good idea to use the RW command instead of rewinding the tape manually by pushing the buttons on the tape unit because the system checks if a job or task is still using the drive before trying to unload the tape. Using the RW command prevents someone from accidentally interrupting a job that is using a tape.

The RW command locks the drive to prevent another job or task from using the tape if it is a scratch tape. You can make the drive available for use again with the RY (Ready) system command, as described in "Readying a Device (RY)" in this section.

Note: *You can set the AUTOUNLOAD option of the MODE (Unit Mode) system command to instruct the operating system to automatically unload tapes from a unit after a reel switch or file close releases the unit. See the* System Commands Reference Manual *for more information on the MODE command.*

Procedure

To rewind a tape from the ODT, enter

```
RW MT <unit number>
```

The unit number is the number of the tape drive with the tape you want to rewind.

If a job or task is presently using the drive, the system displays the message UNIT IN USE.

Example

If you want to rewind the tape on tape drive 118, you enter

```
RW MT 118
```

If the drive is not in use, the system rewinds the tape, unloads and locks the unit, and puts the following in the system messages:

```
MT118 REWOUND AND LOCKED
```

Time and Date

You can display the current system time and date with the TD (Time and Date) system command. You can also set the time, the date, or the time and date on your system, forward or backward. If you change the time forward more than 15 minutes, or the date forward more than three days, or if you change the time or date backward, the system creates a waiting message requesting confirmation. Changing the system time can cause logging, auditing and billing problems.

You can change the system date with the DR (Date Reset) system command. You can change the system time or both the time and date with the TR (Time Reset) system command. Procedures for changing the date, the time, and the time and date follow.

Displaying the Time and Date (TD)

Use the TD (Time and Date) system command to display the current system time, date, and time zone settings.

Procedure

To display the current system time, date, and time zone settings, enter the following command:

```
TD
```

Example

The TD command display looks similar to the following:

```
The date is Thursday August 2, 1992 (92214)

The time is 15:51:59 Pacific Daylight Time (PDT)
```

The number in parentheses following the date on the first line of the display is the Julian date. The Julian date has the form *yyddd,* where *yy* is the year and *ddd* is the day of the year. So Thursday, August 02 is the 214 day of the year 1992.

Changing the Date (DR)

Use the system command DR (Date Reset) to change the system date.

Procedure

To change the system date, use the following syntax:

```
DR <day of week> <day of month> <month> <year>
```

For example, to change the system time to Monday, May 15, 1992, enter

```
DR MONDAY 15 MAY 1992
```

Changing the Date Forward

If the new date is more than three days forward, the system creates a waiting entry indicating the number of days forward. You must enter either a DS (Discontinue) command to cancel the request or an OK (Reactivate) command to confirm it. For example, if you changed the date five days forward, you would get a waiting entry similar to the following:

```
---Job-Task-Pri-Elapsed---------- 1 WAITING ENTRY --------------------
  3367/      5Ø     :54 JOB CHANGEDATE
      VERIFY DATE CHANGE--FORWARD 5 DAYS--OK OR DS
```

To confirm the date change for this example, you would note the job number for the entry (3367, in this case), and then enter

```
3367 OK
```

To discontinue the date change for this example, enter

```
3367 DS
```

If you confirm the date change, the system changes to the new date immediately.

If your request is no more than three days forward, the system changes the date without waiting for confirmation.

Changing the Date Backward

Any request to change the date backward causes the system to generate a waiting entry asking for confirmation. A request to change the clock backward 30 days or more generates two requests for confirmation. You must either confirm these requests with the OK (Reactivate) command or stop the date change with the DS (Discontinue) command.

For example, if you enter a request to change the date backward 30 days, the system responds with a waiting entry similar to the following:

```
---Job-Task-Pri-Elapsed----------- 1 WAITING ENTRY ---------------------
  3367/      8Ø    :14  JOB CHANGEDATE
      CLOCK BEING SET BACKWARDS--CAN CAUSE LOG/AUDIT PROBLEMS--OK OR DS
```

To confirm the request, you enter

```
   3367 OK
```

Notice that the number 3367 is the job number for the waiting entry. Since this request is for 30 days, the system generates a second waiting entry:

```
---Job-Task-Pri-Elapsed----------- 1 WAITING ENTRY ---------------------
  3367/      8Ø    :Ø4  JOB CHANGEDATE
      VERIFY DATE CHANGE--BACKWARD 3Ø DAYS--OK OR DS
```

If you confirm this message, the system date is changed immediately. If the request was for less than 30 days, the date is changed immediately after you confirm the first message.

Changing the Time (TR)

The TR (Time Reset) system command enables you to change just the time, or both the time and date. You can also use the TR command to set the system time zone.

Procedure for Changing the Time

To change just the time, enter the following at the ODT:

```
TR <number>:<number>
```

For example, to change the system time to 11:31, enter

```
TR 11:31
```

You can specify seconds, for example, TR 11:31:30. This command uses a 24-hour clock unless you specify AM or PM. Entering *TR 15:30* is equivalent to entering *TR 3:30 PM*. AM includes midnight; PM includes noon.

If you request a change forward of less than 15 minutes, the change is made immediately. A request for 15 minutes or more forward generates a waiting entry requesting confirmation. Any request to change the time backward generates a waiting entry requesting confirmation.

Procedure for Changing the Time and Date

You can change the system time and date at the same time with the TR command. The procedure is exactly the same as described in the previous procedure, but you use the following syntax:

```
TR <number>:<number> <day of week> <month> <day of month> <year>
```

A request to change the time forward 15 minutes or more, or a request to change the time backward generates a waiting entry requesting confirmation.

Examples

Suppose you want to change the system time to 3:45 p.m., and the system time is now 2:01:10 PM. You enter

```
TR 3:45 PM
```

Since this request is for more than 15 minutes, the system generates a waiting entry:

```
---Job-Task-Pri-Elapsed----------- 1 WAITING ENTRY ---------------------
  3380/      80      :37 JOB CHANGETIME
      VERIFY 1 HOUR 43 MINUTES 50 SECONDS FORWARD
```

This is what you want to do, so you enter

```
3380 OK
```

The time is changed immediately after your confirmation. You can verify this with the system command TD (Time and Date).

Or suppose you want to change the system time and date to 2:30 p.m. Wednesday, June 1, 1992. You enter

```
TR 2:30 PM WEDNESDAY JUN 1 1992
```

Procedure for Changing the System Time Zone

If your site is connected to a network that crosses time zone barriers, having system time stamps from differing time zones can be confusing. To solve this problem, use the TR command with the TIMEZONE option to set the time zone for your system.

Having time zones set for the systems in a network provides the systems with a common time reference. This reference is universal time (UT), which is also known as Greenwich mean time. Your system's time zone is an offset from UT.

The following information tells you how to check your system's current time zone setting, how to remove any time zone setting, and how to specify a time zone for your system. For an extensive explanation of the TR TIMEZONE option, and the available time zone settings, see the *System Commands Reference Manual.*

- To check your system's current time zone setting (if any), enter the following command:

  ```
  TD
  ```

  The time zone is displayed on the second line of the display (for example, "Pacific Daylight Time").

- To remove any time zone setting from your system, enter the following syntax:

  ```
  TR TIMEZONE NONE
  ```

- To specify a time zone for your system, enter the following command:

  ```
  TR TIMEZONE <time zone specification>
  ```

  For a list of the valid predefined time zone specifications, and for an explanation of how to create a custom time zone specification, see the *System Commands Reference Manual.*

Examples

Suppose you want to query your system's current time zone setting, and then set the time zone to central standard time. To find out the current setting, enter

```
TD
```

The system responds with the following display:

```
THE DATE IS THURSDAY AUGUST Ø2, 1992 (92214)

THE TIME IS 15:51:59 Pacific Daylight Time ( PDT )
```

To change the time zone setting to central standard time, you enter the following command:

```
TR TIMEZONE CENTRAL STANDARD
```

USERDATAFILE

All the passwords, usercodes, and usercode attributes are stored in a disk file named SYSTEM/USERDATAFILE. When you want to add, modify, or remove a usercode on your system, you must modify this file with a program called SYSTEM/MAKEUSER.

Modifying the USERDATAFILE with SYSTEM/MAKEUSER (MU)

Instructions on how to use SYSTEM/MAKEUSER are not included in this guide. Only the System Administrator, or the person responsible for adding and changing usercodes should use SYSTEM/MAKEUSER.

You can find detailed instructions for modifying the USERDATAFILE in the *Security Administration Guide.*

Updating and Removing the USERDATAFILE

There will be times when you have to replace the current version of the SYSTEM/USERDATAFILE with an updated version. Because the SYSTEM/USERDATAFILE is a frozen system file, it is protected from being removed or overwritten. To change it, you have to fool the operating system.

To change an active SYSTEM/USERDATAFILE on your system, first use the DL (Disk Location) system command to direct the SYSTEM/MAKEUSER file to another, nonexistent family. Or, you can use the DL command to point to an existing disk family that does not contain the file SYSTEM/MAKEUSER. Either of these methods causes the operating system to issue an RSVP message warning you that either the family you specified or the SYSTEM/MAKEUSER file is non-existant. The system then waits for your response.

While the system waits for your response with an RSVP message, the existing SYSTEM/USERDATAFILE becomes a removable, non-system file. You can remove the existing SYSTEM/USERDATAFILE and copy the new SYSTEM/USERDATAFILE to the same family. After you have successfully copied the USERDATAFILE, cancel the waiting RSVP entry. The USERDATAFILE is once again locked and frozen, and the operating system now accesses the new SYSTEM/USERDATAFILE.

Examples

To remove the active SYSTEM/USERDATAFILE from the file family USERDATA and copy in a new version, enter the following command:

```
DL SYSTEM/USERDATA ON ZZZPACK
```

This command directs the operating system to search for the imaginary pack ZZZPACK. An RSVP message similar to the following appears on the operating console:

```
NO FAMILY ZZZPACK
```

While the operating system waits for a response to the RSVP message, enter the following:

```
REMOVE *SYSTEM/USERDATA FROM USERDATA(PACK)
```

The system responds with the message "SYSTEM/USERDATA removed." Copy the new version of SYSTEM/USERDATA from your tape OPSTAPE to the USERDATA pack.

```
COPY&COMPARE SYSTEM/USERDATA AS *SYSTEM/USERDATA
  FROM OPSTAPE TO USERDATA(PACK)
```

Next, enter the DS command to the RSVP message waiting for ZZZPACK. Your operating system now uses the new SYSTEM/USERDATA file on the USERDATA family.

You can find more extensive information on using the DL system command with the SYSTEM/USERDATAFILE in the *System Commands Reference Manual.*

Glossary

A

active

Pertaining to the state of a process that is executing normally, and is neither scheduled nor suspended.

application software

Programs written to provide specific functions to solve specific problems for end users.

B

backup file

(1) A printer or punch file assigned to a backup peripheral for subsequent output. The default backup peripheral is a disk. (2) A copy of a file that is stored offline so that it can be copied back onto the system if the original file becomes corrupted or inaccessible. (3) A copy of a file on a cataloging system that has been saved with one of the following Work Flow Language (WFL) statements: *COPY & BACKUP*, *ARCHIVE DIFFERENTIAL*, *ARCHIVE FULL*, *ARCHIVE INCREMENTAL*, or *ARCHIVE ROLLOUT*.

base pack

A pack that contains a copy of the system directory for the family of that pack and is currently being used by the system to identify and access the family.

C

CANDE

See Command and Edit.

central processing unit (CPU)

The computer hardware unit that controls and executes instructions contained in object code files.

central processor subsystem

All of the central processing units (CPUs) on a particular system. The total number of CPUs varies from one A Series model to another.

checkerboarding

A situation in which only small areas are available between the in-use areas of disk storage. Many areas can be available, but the system might not be able to use them because none of the areas is large enough.

checksum

(1) A directory test that performs a data integrity check of a directory record to ensure that it has not been corrupted. (2) In Data Management System II (DMSII), a value used to detect certain classes of I/O errors. A checksum is computed for each database file block by applying an equivalence operator to each word in the block. When the block is physically written, the checksum value is stored in a checksum word appended to the end of the block. When the block is read, the checksum is recomputed and the result is compared to the stored value. A checksum error occurs if the two values are not equal.

cold-start

A procedure that can be used during system initialization that places the operating system on a disk and in memory. The system configuration is also defined to the operating system at this time. A cold-start reinitializes all data structures on a disk, causing any information about existing files on that disk to be lost.

Command and Edit (CANDE)

A time-sharing message control system (MCS) that enables a user to create and edit files, and to develop, test, and execute programs, interactively.

Communications Management System (COMS)

A general message control system (MCS) that controls online environments on A Series systems. COMS can support the processing of multiprogram transactions, single-station remote files, and multistation remote files.

compiler

A computer program that translates instructions written in a source language, such as COBOL or ALGOL, into machine-executable object code.

COMS

See Communications Management System.

conditioned tape

A tape that contains one or more keyed files. An attempt to read a conditioned tape shows only nonkeyed files or files for which the license keys are available in the system keys file.

configuration file

(1) A table that contains the configuration of a system. The configuration table is stored in the disk directory of the halt/load family. (2) For the SYSTEM/CONFIGURATOR utility, a file that lists and describes the hardware resources and selected software information that make up a configuration. The configuration file can contain descriptions of several different hardware and software configurations for a system. (3) In the Communications Management System (COMS), a file that contains descriptions of the tables defined through the COMS Utility program. These tables contain information on message routing, security, dynamic program control, and synchronized recovery. This file is also referred to as the COMS CFILE.

continuation pack

A disk that is not currently being used as the base pack for a multipack family. A continuation pack can have a copy of the system directory for the family.

CPU
See central processor unit.

D

data comm
See data communications.

data communications (data comm)
The transfer of data between a data source and a data sink (two computers, or a computer and a terminal) by way of one or more data links, according to appropriate protocols.

data communications processor (DCP)
A hardware component that was replaced by the network support processor (NSP).

data communications data link processor (DCDLP)
A data communications processor that combines the functions of a network support processor (NSP) and a line support processor (LSP) into one physical data link processor (DLP) and supports up to four lines of communication.

data link
In X.25, any serial data communications transmission path, generally between two adjacent nodes or devices and without any intermediate switching nodes.

data link processor (DLP)
A processor that serves as the system interface to a specific peripheral device, controller, or communications network.

data sink
In data communications, that part of the data terminal equipment (DTE) that receives data from a data link.

database (DB)
An integrated, centralized system of data files and program utilities designed to support an application. The data sets and associated index structures are defined by a single description. Ideally, all the permanent data pertinent to a particular application resides in a single database. The database is considered a global entity that several applications can access and update concurrently.

database stack (DBS)
A stack that contains all the information necessary for the Data Management System II (DMSII) Accessroutines to manage a database.

DB
See database.

DBS
See database stack.

DCDLP

See data communications data link processor.

device

Any piece of I/O hardware, such as a data link processor (DLP) or a peripheral.

directory

(1)A table of contents listing the files contained on a device. The device is usually a disk or tape. (2) A list of file names organized into a hierarchy according to similarities in their names. File names are grouped in a directory if their first name constants (and associated usercodes) are identical. These groups are divided into subdirectories consisting of those file names whose first two name constants are identical, and so on.

disk

A random-access data storage device consisting of one or more circular platters that contain information recorded in concentric circular paths called tracks. Data on a disk are accessed by movable read/write heads. Some disks are removable. Synonym for pack, disk pack.

disk drive

The device on which a disk is mounted. The disk drive has movable read/write heads that access the data on the disk.

disk file

A file stored on a disk or disk pack.

DLP

See data link processor.

E

end of file (EOF)

A code at the end of a data file that signals that the last record in the file has been processed.

end of job (EOJ)

The termination of processing of a job.

EOF

See end of file.

EOJ

See end of job.

F

family

(1) One or more disks logically grouped and treated as a single entity by the system. Each family has a name, and all disks in the family must have been entered into the

family with the RC (Reconfigure Disk) system command. (2) The name of the disk or disk pack on which a physical file is located.

family index

A 3-digit number the system assigns to a disk when the disk is added to a family. This family index value must be in the range 1 to 255, inclusive. The base pack is assigned the family index number, the first continuation pack is assigned 002, and so on. A family index is also referred to as a family index number.

family member

A disk that is a base or continuation pack in a family.

family name

(1) The name, consisting of up to 17 alphanumeric characters, assigned by an installation to identify a family of disks. (2) The name (label) of the disk or disk pack on which a physical file is located. The family name of a file is determined by the value of the FAMILYNAME file attribute. (3) The name of the logical group of disk packs on which a physical file is located. A family name consists of from 1 to 17 alphanumeric characters and is assigned by the installation.

family rebuild

The process in which the system reconstructs the file access structure table (FAST) entry for a family by reading its flat directory.

family substitution

A method for redirecting references to files on a disk family to avoid entering the actual family name in commands or file names. For example, if a user enters *FAMILY DISK=PACK OTHERWISE DISK*, that user's file requests are checked on disk packs named PACK and DISK.

FAST

See file access structure table.

field

An area on a screen or form in which data is displayed or entered. The delimiters of the field can be visible or invisible to the terminal operator.

file

A named group of related records. *See* logical file, physical file.

file access structure table (FAST)

Special records in the access structure or catalog directory that the system uses to locate disk files. The FAST contains a pointer to the header of each disk file in the system directory of each family.

file attribute

An element that describes a characteristic of a file and provides information the system needs to handle the file. Examples of file attributes are the file title, record size, number of areas, and date of creation. For disk files, permanent file attribute values are stored in the disk file header.

file equation

A mechanism for specifying the values of file attributes when a program is compiled or executed. A file equation implicitly assigns a value to the FILECARDS task attribute.

file name

(1) A name or word that designates a set of data items. (2) A unique identifier for a file, consisting of name constants separated by slashes. Each name constant consists of letters, digits, and selected special characters. A file name can be optionally preceded by an asterisk (*) or usercode, and optionally followed by ON and a family name.

file title

The complete identifier for a file that consists of the file name, the word ON, and the family name.

file transfer

The communication of files between host systems, workstations, or terminals. The category of distributed systems service (DSS) that enables a user to transfer files between host systems, workstations, or terminals. *See also* library maintenance.

flat directory

See system directory.

frozen

The state of a process whose STATUS task attribute has the value FROZEN. This STATUS value indicates that the process is a library process and provides objects that can be imported by user processes.

G

GCR

See group-coded recording.

group-coded recording (GCR)

A scheme for recording data on a magnetic tape (MT).

guard file

A disk file created by the GUARDFILE utility program that describes the access rights of various users and programs to a program, data file, or database.

H

halt/load

A system-initialization procedure that temporarily halts the system and loads the master control program (MCP) from a disk to main memory.

halt/load family

The disk family that contains the currently operative master control program (MCP) object code file.

halt/load pack

A pack that contains a master control program (MCP) object code file designated as the currently operative MCP. This pack is mounted on a drive that has been or will be designated as the halt/load unit.

halt/load unit

The disk pack that contains the currently operative master control program (MCP) object code file.

header

(1) A data structure that contains information about a disk file, such as the physical location of the file on the disk and various file attributes. A header is also referred to as a disk file header. (2) A sequence of characters preceding the text of a message, containing routing or other communications-related information.

home

The uppermost, leftmost input character position of a screen or form.

I

I/O processor (IOP)

A specialized processor for moving data between system memory and the I/O subsystem.

I/O subsystem

The hardware and software that manage all transfers of information between the operating system and peripheral devices.

IDC

See Interactive Datacomm Configurator.

independent runner (IR)

A master control program (MCP) procedure that is initiated as an independent process. The procedure is executed in its own process stack rather than in the stack of a user process. An independent runner (IR) can be either visible or invisible. If the IR is visible, its status can be interrogated. If the IR is invisible, it does not appear in mix displays.

initialization

(1) The process of starting a program and giving starting values to variables.
(2) A procedure that makes a system or subsystem available for its intended use. Important phases of initialization are the recognition of the physical environment, the identification of the available resources, and the establishment of the interface with the user. System initialization occurs as part of a halt/load.

Interactive Datacomm Configurator (IDC)

A Unisys interactive, menu-driven utility that enables the user to create, interrogate, and modify data communications network configurations.

internal file name

The name used to declare a logical file in a program. The internal name of a file is given by the value of its INTNAME file attribute. Work Flow Language (WFL) file equation statements can reference the file by implicitly or explicitly specifying an INTNAME value that matches the INTNAME attribute of a file in a program.

IOP

See I/O processor.

IR

See independent runner.

J

job

An independent process. The job of a particular task is the independent process that is the eldest ancestor of that task.

job file

A disk file that is associated with a job and contains the job log. The job file for a Work Flow Language (WFL) job also serves as the object code file for the job, and includes job restart information, data specifications, and a copy of the WFL source program.

job queue

A structure in the system software that stores a list of jobs that have been compiled and are waiting to be initiated.

job queue number

A unique identifier for a particular job queue.

L

label

(1) The first 28 sectors on a disk, on which information about the disk is stored. This information includes the family name and serial number, the Master Available Table (MAT), the family index number, information about the family base pack, and a pointer to the system directory if the disk contains a directory. (2) An area on a magnetic tape (MT) that contains permanent attributes associated with the tape volume or with individual files on the volume, such as the volume serial number and the file name. (3) In some programming languages, a name that identifies either a point in the Calculation Specifications where a GOTO operation branches or the beginning of a subroutine.

library

(1) A collection of one or more named routines or library objects that are stored in a file and can be accessed by other programs. (2) A program that exports objects for use by user programs. (3) A collection of related files.

library maintenance

A master control program (MCP) procedure that copies disk files to and from disk, tape, and compact disk (CD). Library maintenance is invoked by the Work Flow Language (WFL) statements *ADD, COPY,* and *ARCHIVE. See also* file transfer.

library maintenance tape

A tape created by library maintenance that contains backup copies of disk files.

line support processor (LSP)

The data communications subsystem processor that manages communication with the host and initiates processes that control the input of messages to and the output of messages from data communications lines.

logical file

A file variable declared in a program, which represents the file and its structure to the program. A logical file has no properties of its own until it is described by file attributes or associated with a physical file.

LSP

See line support processor.

M

maintenance display terminal (MDT)

ON A 1 through A 6 systems, the name given to the System Control Terminal (SCT) when it is in maintenance mode. The MDT enables the operator to access the maintenance subsystem.

maintenance subsystem

The software and hardware that serve as the interface between the user and the hardware. This software can initialize or configure the hardware, and halt or initialize system software. The subsystem is also used in diagnosing hardware and software problems.

MARC

See Menu-Assisted Resource Control.

master control program (MCP)

The central program of the A Series operating system. The term applies to any master control program that Unisys might release for A Series systems.

MCP

See master control program.

MCP/AS

See Master Control Program/Advanced Systems.

MCS

See message control system.

MDT

See maintenance display terminal.

memory

A temporary storage area where data and programs are placed while they are being processed.

Menu Assisted Resource Control (MARC)

A menu-driven interface to A Series systems that also enables direct entry of commands.

message control system (MCS)

A program that controls the flow of messages between terminals, application programs, and the operating system. MCS functions can include message routing, access control, audit and recovery, system management, and message formatting.

mix

The set of processes that currently exist on a particular computer. The mix can include active, scheduled, and suspended processes.

mix number

A 4-digit number that identifies a process while it is executing. This number is stored in the MIXNUMBER task attribute.

N

Network Information File II (NIFII)

The file generated when a Network Definition Language II (NDLII) program is compiled. This file contains line support processor (LSP) and network support processor (NSP) code, data structures, and other information. A NIFII is also generally referred to as a network information file (NIF).

network support processor (NSP)

A data communications subsystem processor that controls the interface between a host system and the data communications peripherals. The NSP executes the code generated by the Network Definition Language II (NDLII) compiler for line control and editor procedures. An NSP can also control Line Support Processors (LSPs).

NIF

See Network Information File II.

NIFII

See Network Information File II.

NSP

See network support processor.

O

object code

The instructions in machine code that are created as a result of compiling source code.

ODT

See operator display terminal.

offspring

The dependent process whose critical block is owned by a particular parent process.

operating system

The set of programs that control the operational environment of a computer system by activities such as managing processors, memory, and peripherals, logging system activities, enforcing security, and executing system commands. On A Series systems, the operating system consists of a master control program (MCP) and system libraries such as CENTRALSUPPORT, GENERALSUPPORT, JOBFORMATTER, and PRINTSUPPORT.

operator display terminal (ODT)

(1) A system control terminal (SCT) configured for direct communication with the operating system. The ODT is used primarily by operations personnel for entering commands that control and direct the system and its resources. (2) The name given to the system control terminal (SCT) when it is used as an ODT.

P

pack (PK)

A random-access data storage device consisting of one or more circular platters that contain information recorded in concentric circular paths called tracks. Data on a pack are accessed by movable read/write heads. Some packs are removable.

parent

A process that owns the critical block of a dependent process. If the parent exits the critical block before the dependent process terminates, the dependent process is discontinued.

PE

See phase encoded.

peripheral

A device used for input, output, or file storage. Examples are magnetic tape drives, disk drives, printers, or operator display terminals (ODTs). *Synonym for* peripheral device.

peripheral device

A device used for input, output, or file storage. Examples are magnetic tape drives, printers, disk drives, and operator display terminals (ODTs). *Synonym for* peripheral.

phase encoded (PE)

A scheme for recording data on a magnetic tape (MT) in which the phase of the carrier shifts when the data switches from 0 to 1 or from 1 to 0.

physical file

A file as it is stored on a particular recording medium such as a disk or a tape.

PIB

See process information block.

PK

See pack.

primitive system command

Any system command that begins with two question marks (??). Primitive system commands bypass the system command handler and can be helpful when the system does not respond to nonprimitive system commands.

print request

(1) A request that contains information describing the location, time, and method of printing a printer file or group of printer files. (2) A group of one or more files to be printed, along with the attributes that describe when, where, and how to print them.

Print System (PrintS)

A Unisys software product used to control when, where, and how printer backup files are printed on A Series systems.

printer backup file

A system-formatted file that contains data to be printed and carriage control information. A printer backup file refers to both printer and card punch output.

PrintS

See Print System.

priority

A characteristic associated with a process that determines its precedence in the use of system resources. A process with higher priority executes more quickly than it would if it had lower priority.

private file

A file with a SECURITYTYPE attribute specified as PRIVATE. Only users logged on to a privileged usercode or to the usercode under which the file is stored can access a private file.

privilege

The ability to invoke actions that are not ordinarily allowed, such as accessing private files stored under other usercodes or invoking privileged functions such as SETSTATUS. The concept of privilege applies to usercodes, programs, and processes.

procedure

A block that can be invoked by statements elsewhere in the same program or, in some cases, by statements in another program. In most instances, a procedure has a procedure heading and a procedure body. Examples are a procedure in ALGOL, a procedure or function in Pascal, a subroutine or function in FORTRAN, or a complete COBOL program.

process

(1) The execution of a program or of a procedure that was initiated. The process has its own process stack and process information block (PIB). It also has a code segment dictionary, which can be shared with other processes that are executions of the same program or procedure. (2) A software application; that is, any activity or systematic sequence of operations that produces a specified result.

process information block (PIB)

A memory structure that is associated with each process stack and code segment dictionary. The PIB contains control information that is visible only to the operating system. The PIB for a process stack also contains a reference to a task attribute block (TAB).

processor

A hardware component that executes programs and procedures. *See also* central processing unit.

public file

A file with a SECURITYTYPE attribute specified as PUBLIC. Users who are logged on to any usercode can access a public file by specifying the (<usercode>)<file name> form for the file title.

Q

queue

A data structure used for storing objects; the objects are removed in the same order they are stored. *See also* job queue, ready queue.

R

RDT

See remote display terminal.

ready queue (READYQ)

A list, maintained by the operating system, of the processes that are waiting for service from a processor.

remote display terminal (RDT)

On A 1 through A 6 systems, The name given to the system control terminal (SCT) when it is in remote display mode. The RDT enables the operator to access remote support.

Remote Print System (ReprintS)

A Unisys software system that controls the routing and printing of backup files at remote (data comm) destinations and on BNA networks.

ReprintS

See Remote Print System.

RSVP message

A message the system displays for a suspended process that states the reason the process was suspended. RSVP messages ask for a reply such as *OK* or *DS*.

S

scheduled process

A process whose initiation is delayed, either because the operator has entered an HS (Hold Schedule) system command or because the operating system estimates the process is likely to need more memory than is currently available.

scratch tape

A labeled magnetic tape (MT) whose label indicates that there are no files on the tape. Old data might remain on the tape, but this old data cannot be read unless the tape is read as an unlabeled tape. The old data present on a scratch tape is written over when new data is written to the tape.

SCT

See system control terminal.

sector

A subdivision of a track on a disk. A sector is the minimum addressable area on a disk pack. Unisys A Series system sectors are 30 words, or 180 bytes, long.

segment

Synonym for sector.

source code

An instruction or a set of instructions written in a programming language. Source code must be translated (compiled) to object code before the program can be executed.

spooling

The process of indirectly sending output files to peripheral devices such as printers. The output file is logically written to the peripheral device, but actually is sent to a tape or disk file known as a backup file. The backup file is written to the peripheral device when the job finishes. Spooling enables many users to share peripherals efficiently by preventing any one job from monopolizing a peripheral.

subdirectory

See directory.

subsystem

A secondary or subordinate system usually capable of operating independently of, or asynchronously with, a controlling system.

support library

A library that is associated with a function name. User programs can access a support library by way of its function name instead of its object code file title. The operator uses the SL (Support Library) system command to link function names with libraries.

system command

Any of a set of commands used to communicate with the operating system. System commands can be entered at an operator display terminal (ODT), in a Menu-Assisted Resource Control (MARC) session, or by way of the DCKEYIN function in a privileged Data Communications ALGOL (DCALGOL) program.

system control terminal (SCT)

A terminal used to enter information. An SCT can be used three ways: as an operator display terminal (ODT) to interface with the operating system, as a maintenance display terminal (MDT) to interface with the maintenance subsystem, or as a remote display terminal (RDT) to interface with remote support. The windows providing these uses are available once the automatic initialization sequence has finished.

system directory

(1) A special disk file on each disk family that the system uses to locate files on that family. The system directory, also referred to as the flat directory, contains a disk file header for each permanent file in the family. (2) The logical directory for nonusercoded files.

system file

A file with a special security status that protects it from being removed, retitled, or replaced except by selected system interfaces. For example, the job description (JOBDESC) file can be removed only by the ??RJ (Remove JOBDESC File) system command.

system library

A library that is part of the system software and is accorded special privileges by the operating system. Two examples of system libraries are GENERALSUPPORT and PRINTSUPPORT.

T

tape drive

An I/O peripheral device that stores data on reels or cartridges of magnetic tape (MT).

task

(1) A dependent process. (2) Any process, whether dependent or independent. *See also* process.

task attribute

Any of a number of items that describe and control various aspects of process execution such as the usercode, priority, and the default family specification. Task attributes can be assigned interactively through task equations, or programmatically through statements that use task variables.

task equation

A mechanism for specifying the values of task attributes when a program is compiled or executed. A task equation can consist of task attribute assignments, file equations, and database equations.

U

usercode

An identification code used to establish user identity and control security, and to provide for segregation of files. Usercodes can be applied to every task, job, session, and file on the system. A valid usercode is identified by an entry in the USERDATAFILE.

utility

A system software program that performs commonly used functions.

V

variable

(1) An object in a program whose value can be changed during program execution.
(2) In the Screen Design Facility Plus (SDF Plus), a component of a form that stores data entered in the fields of the form image or the return value for a menu or a function key. A variable is also referred to as a display variable. An item that appears in syntax notation as a variable item.

W

WFL

See Work Flow Language.

WFL job

(1) A Work Flow Language (WFL) program, or the execution of such a program. (2) A collection of Work Flow Language (WFL) statements that enable the user to run programs or tasks.

Work Flow Language (WFL)

A Unisys language used for constructing jobs that compile or run programs on A Series systems. WFL includes variables, expressions, and flow-of-control statements that offer the programmer a wide range of capabilities with regard to task control.

Work Flow Language (WFL) commands

Commands interpreted by the WFL compiler.

Bibliography

A Series CANDE Configuration Reference Manual (8600 1344) Unisys Corporation.

A Series CANDE Operations Reference Manual (8600 1500). Unisys Corporation.

A Series Communications Management System (COMS) Operations Guide (8600 0833) Unisys Corporation.

A Series Disk Subsystem Administration and Operations Guide (8600 0668). Unisys Corporation.

A Series Documentation Library Overview (8600 0361). Unisys Corporation.

A Series GETSTATUS/SETSTATUS Programming Reference Manual (8600 0346). Unisys Corporation.

A Series I/O Subsystem Programming Guide (8600 0056). Unisys Corporation. Formerly *A Series I/O Subsystem Programming Reference Manual.*

A Series Interactive Datacomm Configurator (IDC) Operations Guide (1169810). Unisys Corporation.

A Series Memory Subsystem Administration and Operations Guide (1169836). Unisys Corporation.

A Series Menu-Assisted Resource Control (MARC) Operations Guide (8600 0403). Unisys Corporation.

A Series Operating System Installation Guide (8600 1021). Unisys Corporation.

A Series Print System (PrintS/ReprintS) Administration, Operations, and Programming Guide (8600 1039). Unisys Corporation.

A Series Printing Utilities Operations Guide (8600 0692). Unisys Corporation.

A Series Security Administration Guide (8600 0973). Unisys Corporation.

A Series Security Features Operations and Programming Guide (8600 0528). Unisys Corporation.

A Series Software Documents Master Index (8600 1542). Unisys Corporation.

A Series Software Release Installation Guide (8600 0981). Unisys Corporation.

A Series System Administration Guide (8600 0437). Unisys Corporation.

A Series System Commands Operations Reference Manual (8600 0395). Unisys Corporation.

A Series System Messages Support Reference Manual (8600 0429). Unisys Corporation.

A Series System Software Support Reference Manual (8600 0478). Unisys Corporation.

A Series System Software Utilities Operations Reference Manual (8600 0460). Unisys Corporation.

A Series Systems Functional Overview (8600 0353). Unisys Corporation.

A Series Task Attributes Programming Reference Manual (8600 0502). Unisys Corporation. Formerly *A Series Work Flow Administration and Programming Guide.*

A Series Work Flow Language (WFL) Programming Reference Manual (8600 1047). Unisys Corporation.

Index

A

D

E

F

G

H

I

J

K

N

O

P

Q

R

S

T

U

V

W

Y

Z

86000387-100