Executive

Convergent Technologies

EXECUTIVE MANUAL

Specifications Subject to Change.

Third Edition (November 1983) A-Ø9-ØØ33Ø-Ø1-A

CONTENTS

APPENDIXES

LIST OF FIGURES

LIST OF TABLES

GUIDE TO TECHNICAL DOCUMENTATION

This manual is one of a set that documents the Convergent™ Family of Information Processing Systems. The set can be grouped as follows:

Introductory
- Installation Guide
- Operator's Guide
- Executive Manual
- Context Manager
- Status Codes Manual
- Installation Guide (NGEN)
- Operator's Guide (NGEN)

Hardware
- NGEN
  - Processor Manual: Model CP-001
  - Dual Floppy Disk Manual
  - Floppy/Hard Disk Manual
  - Diagnostics Manual
  - Keyboard Manual
  - Power System Manual
  - Monochrome Monitor Manual: Model VM-001
  - Color Monitor Manual
- IWS
  - Workstation Hardware Manual
  - Peripherals Hardware Manual
  - IWS Peripherals Hardware Manual (SMD Version)
- AWS
  - AWS-210 Hardware Manual
  - AWS-220, -230, -240 Hardware Manual
  - AWS Color Workstation Hardware Manual

Operating System
- CTOS™ Operating System Manual
- System Programmer's Guide

Guest Operating Systems
- CP/M-86™
- MS™-DOS (and GW™-BASIC)
- XENIX™

Programming Languages
- COBOL Manual
- FORTRAN Manual
- FORTRAN-86 Manual
- BASIC Manual
- BASIC Compiler Manual
- Pascal Manual
- Assembly Language Manual

Program Development Tools

- COBOL Animator
- Editor Manual
- Debugger Manual
- Linker/Librarian Manual

Data Management Facilities

- CT-DBMS™ Manual
- ISAM Manual
- Forms Manual
- Sort/Merge Manual

Text Management Facilities

- Word Processing User's Guide
- Word Processing Reference Manual
- Word Processing Quick Reference

Applications Facilities

- Project Planner Manual
- CT-MAIL™ User's Reference Manual
- CT-MAIL™ Administrator's Reference Manual
- Multiplan
- Business Graphics User's Guide
- Business Graphics Reference Manual
- Graphics Programmer's Guide
- Font Designer Manual

Communications

- Asynchronous Terminal Emulator Manual
- 3270 Terminal Emulator Manual
- 2780/3780 RJE Terminal Emulator Manual
- SNA Network Gateway Manual
- SNA 3270 Emulator Manual
- X.25 Network Gateway Manual
- Multimode Terminal Emulator User's Guide
- Multimode Terminal Emulator Reference Manual

This section outlines the contents of these manuals.

INTRODUCTORY

The Installation Guide describes the procedure for unpacking, cabling, and powering up a system.

The Operator's Guide addresses the needs of the average user for operating instructions. It describes the workstation switches and controls, keyboard function, and floppy disk handling.

The Executive Manual describes the command interpreter, the program that first interacts with the user when the system is turned on. It describes available commands and discusses command execution, file management, program invocation, and system management. It also addresses status inquiry, volume management, the printer spooler, and execution of batch jobs. This manual now incorporates the System Utilities and Batch Manuals.

The Context Manager Manual describes and teaches the use of the Context Manager, which allows the user to run applications concurrently and interchange them on the screen almost instantly.

The Status Codes Manual contains complete listings of all status codes, bootstrap ROM error codes, and CTOS initialization codes. The codes are listed numerically along with any message and an explanation.

The NGEN Installation Guide describes the procedure for unpacking, assembling, cabling, and powering up an NGEN workstation.

The NGEN Operator's Guide is a link between the operator, the NGEN workstation, and the workstation's documentation. The Operator's Guide describes the operator controls and the use of the floppy disk drives, as well as how to verify that the workstation is operational and how to use software release notices.

HARDWARE

NGEN

The Processor Manual: Model CP-001 describes the Processor Module, which houses the Processor board, Memory board, I/O board, Video/Keyboard board, and Motherboard. It details the architecture and theory of operations of the printed circuit boards, external interfaces, and the Memory Expansion Cartridge, as well as the X-Bus specifications.

The Dual Floppy Disk Manual and the Floppy/Hard Disk Manual describe the architecture and theory of operation for the NGEN modules. They discuss the respective disk drives and controllers, and contain the applicable OEM disk drive manuals.

The Diagnostics Manual describes the diagnostics available for the NGEN workstation. It discusses the Processor Module's bootstrap ROM program and error codes, and individual software diagnostics for modules in the workstation.

The Keyboard Manual describes the theory of operation for the NGEN keyboard.

The Power System Manual describes the operation and connections for the 36-Volt Power Supply and the dc/dc converters used with the NGEN workstation.

The Monochrome Monitor Manual: Model VM-001 describes the operation and connections of the 12-inch Monochrome Monitor used with the NGEN workstation.

The Color Monitor Manual describes the operation and connections of the 15-inch Color Monitor used with the NGEN workstation.

IWS

The Workstation Hardware Manual describes the mainframe, keyboard, and video display for the IWS family of workstations. It specifies system architecture, printed circuit boards (Motherboard, Processor, I/O Memory, Multiline Communications Processor, Video Control, Graphics Control Board, ROM and RAM Expansions), keyboard, video monitor, Multibus interface, communications interfaces, power supply, and environmental characteristics of the workstation.

The Peripherals Hardware Manual describes the non-SMD single-board Mass Storage Subsystem (MSS) and Mass Storage Expansion (MSX) disk subsystems for the IWS family of workstations. It contains descriptions of the disk controller Motherboard, the two controller boards for floppy and Winchester disks, power supplies, disk drives, and environmental characteristics.

The IWS Peripherals Hardware Manual (SMD Version) describes the SMD MSS and MSX disk subsystems having one controller board.

AWS

The AWS-210 Hardware Manual describes the mainframe, keyboard, and video display of the AWS-210 workstation. It specifies architecture, theory of operation of the printed circuit boards (Motherboard, Deflection, and CPU), keyboard, video monitor, expansion interface, cluster communications interface, power supply, and environmental characteristics of the workstation.

The AWS-220, -230, -240 Hardware Manual describes the mainframe, keyboard, disk controllers, and video display of the AWS-220, -230, and -240 workstations. It specifies architecture, theory of operation of the printed circuit boards (Motherboard, Deflection, 8088 CPU, 8086 CPU, Floppy Disk Controller, and Hard Disk Controller), keyboard, video monitor, cluster communications interface, external interfaces, power supply, and environmental characteristics of the workstation.

The AWS Color Workstation Hardware Manual describes the mainframe, keyboard, and color video display of the AWS Color Workstation. This manual reports the architecture and theory of operation of the printed circuit boards (Motherboard, Graphics Control Board, Hard Disk Controller, Color Video, Color Deflection, and CPU), keyboard, color monitor, peripheral interfaces, cluster communications interface, power supply, and environmental characteristics of the workstation. This manual also contains four OEM disk drive manuals and a summary of adjustments for the color monitor.

OPERATING SYSTEM

The CTOS™ Operating System Manual describes the operating system. It specifies services for managing processes, messages, memory, exchanges, tasks, video, disk, keyboard, printer, timer, communications, and files. In particular, it specifies the standard file access methods: SAM, the sequential access method; RSAM, the record sequential access method; and DAM, the direct access method.

The System Programmer's Guide addresses the needs of the system programmer or system manager for detailed information on operating system structure and system operation. It describes (1) cluster architecture and operation, (2) procedures for building a customized operating system, and (3) diagnostics.

GUEST OPERATING SYSTEMS

The CP/M-86™ and MS™-DOS Manuals describe the single-user operating systems originally designed for the 8086-based personal computer systems.

The GW™-BASIC Manuals describe the version of BASIC that runs on the MS™-DOS Operating System.

The XENIX™ Manuals describe the 16-bit adaptation of the UNIX system, including the XENIX environment for software development and text processing.

PROGRAMMING LANGUAGES

The COBOL, FORTRAN, FORTRAN-86, BASIC (Interpreter), BASIC Compiler, PASCAL, and Assembly Language Manuals describe the system's programming languages. Each manual specifies both the language itself and also operating instructions for that language.

The Pascal Manual is supplemented by a popular text, Pascal User Manual and Report.

The Assembly Language Manual is supplemented by a text, the Central Processing Unit, which describes the main processor, the 8086. It specifies the machine architecture, instruction set, and programming at the symbolic instruction level.

PROGRAM DEVELOPMENT TOOLS

The COBOL Animator describes the COBOL Animator, a debugger that allows the user to interact directly with the COBOL source code during program execution.

The Editor Manual describes the text editor.

The Debugger Manual describes the Debugger, which is designed for use at the symbolic instruction level. It can be used in debugging FORTRAN, Pascal, and assembly-language programs. (COBOL and BASIC, in contrast, are more conveniently debugged using special facilities described in their respective manuals.)

The *Linker/Librarian Manual* describes the Linker, which links together separately compiled object files, and the Librarian, which builds and manages libraries of object modules.

DATA MANAGEMENT FACILITIES

The *CT-DBMS™ Manual* describes Convergent's data base management system (CT-DBMS), which consists of (1) a data manipulation language for accessing and manipulating the data base and (2) utilities for administering the data base activities such as maintenance, backup and recovery, and status reporting.

The *ISAM Manual* describes both the single- and the multiuser indexed sequential access method. It specifies the procedural interfaces (and how to call them from various languages) and the utilities.

The *Forms Manual* describes the Forms facility that includes (1) the Forms Editor, which is used to interactively design and edit forms, and (2) the Forms run time, which is called from an application program to display forms and accept user input.

The *Sort/Merge Manual* describes (1) the Sort and Merge utilities that run as a subsystem invoked at the Executive command level, and (2) the Sort/Merge object modules that can be called from an application program.

TEXT MANAGEMENT FACILITIES

The *Word Processing User's Guide* introduces the Word Processor to the first-time user. It provides step-by-step lessons that describe basic word processing operations. The lessons show how to execute operations and apply them to sample text.

The Word Processing Reference Manual is a reference tool for users already familiar with the Word Processor. It describes the Word Processor keyboard and screen; basic, advanced, and programmer-specific operations; list processing; printer and print wheel configurations; and hardware considerations.

The Word Processing Quick Reference provides a concise summary of all word processing operations and briefly describes the keyboard and commands.

APPLICATIONS FACILITIES

The Project Planner schedules and analyzes tasks, milestones, and the allocation of resources in a project. By means of diagrams and several kinds of bar charts, Project Planner presents time and resource allocation results and shows the occurrence of project milestones. The Project Planner Manual explains the use of the program and also serves as a reference once the user is familiar with it.

The CT-MAIL™ User's Reference Manual introduces the first-time user to the CT-MAIL electronic mail system. It provides step-by-step instructions for using the basic CT-MAIL operations to create, send, and receive mail.

The CT-MAIL™ Administrator's Reference Manual provides the System Administrator with instructions for installing, configuring, and maintaining the CT-MAIL electronic mail system; setting up communication lines; creating and maintaining mail centers; adding mail users; creating distribution lists; and troubleshooting.

Multiplan is a financial modeling package designed for business planning, analysis, budgeting, and forecasting.

The Business Graphics User's Guide introduces Business Graphics to the first-time user. It provides step-by-step lessons that describe basic Business Graphics operations. The lessons show how to execute operations and apply them to sample charts.

The Business Graphics Reference Manual is a reference tool for users already familiar with

Business Graphics. It describes the Business Graphics keyboard and screen; box and arrow cursor movement; obtaining information from Multiplan; operations; and plotter configurations.

The Graphics Programmer's Guide is a reference for applications and systems programmers. It describes the graphics library procedures that can be called from application systems to generate graphic representations of data, and it includes a section on accessing Business Graphics from an application system.

The Font Designer Manual describes the interactive utility for designing new fonts (character sets) for the video display.

COMMUNICATIONS

The Asynchronous Terminal Emulator Manual describes the asynchronous terminal emulator.

The 3270 Terminal Emulator Manual describes the 3270 emulator package.

The 2780/3780 RJE Terminal Emulator Manual describes the 2780/3780 emulator package.

The SNA Network Gateway Manual describes the SNA Network Gateway, which supports data communications over an SNA network. The SNA Network Gateway comprises the Transport Service and Status Monitor. The Transport Service allows a Convergent workstation to function as cluster controller and forms the foundation for Convergent SNA products.

The SNA 3270 Emulator Manual describes the SNA 3270 emulator package. The SNA 3270 emulator provides CRT and printer subsystems in addition to a Virtual Terminal Interface for use in application programs.

The X.25 Network Gateway Manual describes the X.25 Network Gateway, which supports CCITT Recommendation X.25 communications over a public data network. There are three levels of access to the network: packet, X.25 sequential access method, and the Multimode Terminal Emulator X.25 communications option.

The Multimode Terminal Emulator User's Guide introduces the Multimode Terminal Emulator to the first-time user. It describes the MTE video display, keyboard, display memory, and advanced operations for the X.25 communications option.

The Multimode Terminal Emulator Reference Manual is a reference tool for sophisticated users of the Multimode Terminal Emulator. It describes the MTE escape sequences and field verification program.

CP/M-86 is a trademark of Digital Research.

MS, GW and XENIX are trademarks of Microsoft Corp.

UNIX is a trademark of Bell Laboratories.

1 OVERVIEW

WHAT IS THE EXECUTIVE?

The Executive is an interactive command interpreter that you can use to tell the computer what to do. The Executive acts as an interface between you and the operating system or between you and one of the various application systems installed on your workstation.

The Executive recognizes commands you give it and prompts you for the information it needs to execute the command by displaying a command form for you to complete.

WHAT TASKS CAN THE EXECUTIVE PERFORM?

Commands are available to perform file management tasks, to invoke user programs or applications systems, to query the system for status information, and to perform system management functions.

USING THIS MANUAL

The Executive manual is designed to acquaint both the novice and the experienced user with the capabilities of the Executive. Therefore, within each section, concepts are presented in an order of increasing complexity.

The Executive is very easy to use. You do not need to understand all the concepts presented in this manual to use it effectively.

ORGANIZATION

The first 5 sections act as a User's Guide. Concepts are explained in these sections and instructions on how to use the various Executive features are included.

Section 1, "Overview," gives a brief description of the Executive and this manual.

Section 2, "How to Sign On," explains how to fill out the SignOn form.

Section 3, "Screen and Keyboard," describes the screen, and discusses how to change screen characteristics. The special functions of keyboard keys in the Executive and the use of the type-ahead feature are discussed here.

Section 4, "Basic Concepts," covers the most frequently used features of the Executive. It discusses executing a command, file specification, file management, and file protection.

Section 5, "Advanced Concepts," covers less frequently used features and advanced material for system analysts and system administrators. Advanced command execution techniques, using submit files, using byte streams, invoking user programs, status inquiry, user configuration, system protection, volume management, cluster management, partition management, and printer management are all discussed here.

Section 6, "Commands," describes each Executive command in detail and is arranged alphabetically.

Section 7, "Batch Processing," describes how to use the batch processing facilities and JCL statements.

Tables, examples, and programmer's notes on the spooler appear in the Appendixes.

The Glossary includes many terms that are also defined in the body of the text.

CONVENTIONS USED

New terms are underlined.

The names of all symbols are spelled out. The actual symbol is shown in parentheses following its name. Example: asterisk (*)

Key names are used as they appear on the keyboard.

The SignOn form (shown in Figure 2-1) acts as a gateway to the Executive and the CTOS operating system. The SignOn form is displayed on the screen when the system is turned on and whenever you have logged out.

The SignOn form is displayed by a program called SignOn. Completing the form identifies you as a user to the SignOn program.

The Executive is one of the options your system administrator can arrange to have invoked after SignOn. Other possibilities include the Context Manager or a specific application system, such as the Word Processor.

These options are specified for each user in a file used by the SignOn program called the User Configuration file. Your User Configuration file is set up for you by your system administrator. (For more information on User Configuration files see the subsection "User Configuration" in Section 5, "Advanced Concepts."

SIGNON FORM

The SignOn form has three fields, User name, Password, and Date/Time.

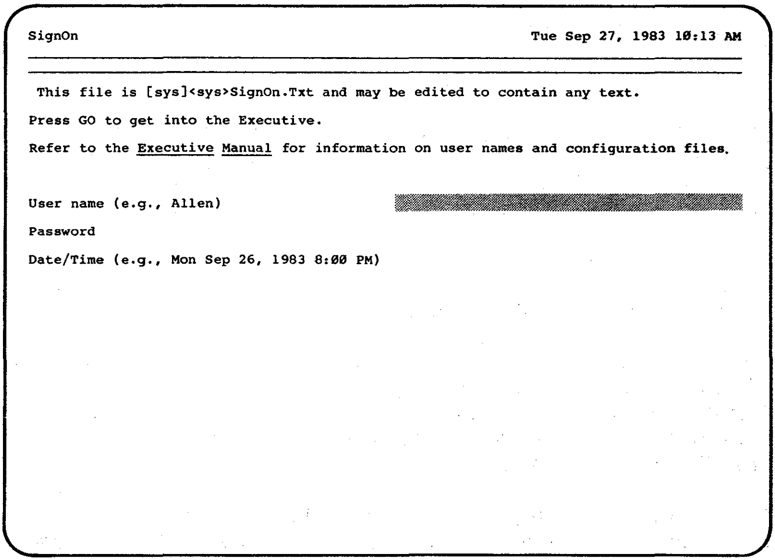
The highlight (shown in Figure 2-1 by a shaded line) indicates the current field. To move from field to field use the RETURN key, the NEXT key, or the up arrow or down arrow key.

COMPLETING THE SIGNON FORM

To sign on, you type required information into each of the three fields of the SignOn form and then press GO.

USER NAME

As a system user you have a unique user name, which identifies you to the system.



```
SignOn                                                Tue Sep 27, 1983 10:13 AM

 This file is [sys]<sys>SignOn.Txt and may be edited to contain any text.
Press GO to get into the Executive.
Refer to the Executive Manual for information on user names and configuration files.

User name (e.g., Allen)
Password
Date/Time (e.g., Mon Sep 26, 1983 8:00 PM)
```

Figure 2-1. SignOn Form.

Your user name is configured for you by your system administrator. Type your user name in this field.

PASSWORD

Enter either your volume password or a file password here. You can leave this field blank if the volume has no password or if you only intend to access unprotected areas.

Access to files within the system is restricted to those files for which the password you entered is applicable.

If you have a password protected volume, you can sign on without a password, if your system has been configured that way, but access to files is restricted. You can enter the password at that point with the Path command.

Password protection is discussed in the subsection "File Protection" in Section 4, "Basic Concepts."

DATE/TIME

This field is optional unless the system clock is not set and no valid date and time can be found on the disk.

The Date/Time information is used to indicate creation and modification dates for files. Setting the time sets the system clock for the entire cluster.

If the system clock has not been set, you must enter the date and time. The date shown in the Status frame at the top of the screen flashes. You may accept this date by pressing GO. To change the flashing date to a current one, type the current date and time into the command field and press GO.

When entering the date and time, you must enter the day of the week, month, date, year and time. You can use the example given on the form as a guide. The date and time information can be entered in almost any format. However, the day of the week and the date given must be a valid combination, since the system verifies the date you give against an internal calendar.

See the Set Time command in Section 6, "Commands."

3 SCREEN AND KEYBOARD

The Executive uses the standard screen and keyboard. Both include some special features.

SCREEN

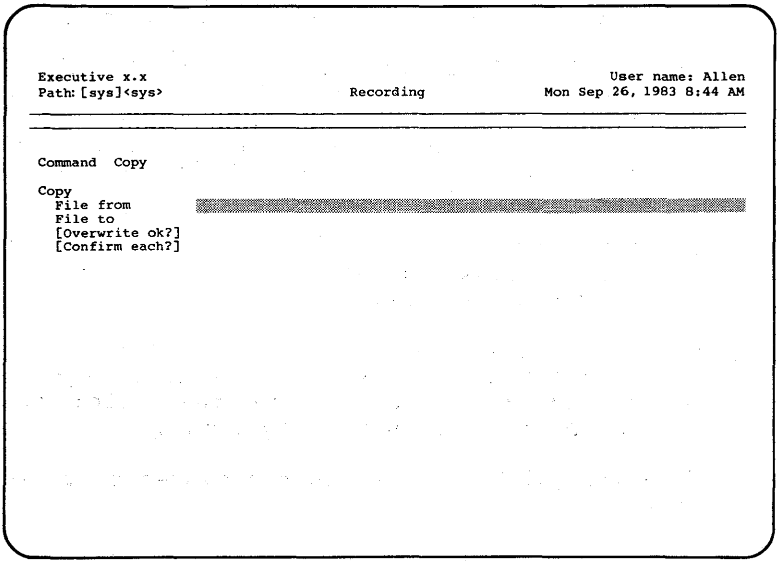
When you have completed the SignOn form and press GO, the Executive screen (shown in Figure 3-1) appears.

The Executive screen is divided into three areas:

- o Status frame

- o Event frame

- o Command frame



```
Executive x.x                                                  User name: Allen
Path: [sys]<sys>                  Recording                Mon Sep 26, 1983 8:44 AM

Command  Copy

Copy
  File from
  File to
  [Overwrite ok?]
  [Confirm each?]
```

Figure 3-1. Executive Screen Diagram.

STATUS FRAME

The top two lines of the screen constantly display important system status information. This area of the screen is called the Status frame. The Status frame is separated from the rest of the screen by a double bar.

From left to right the top line of the Status frame displays

- the name and version of the Executive
- the release number and type of operating system presently running
- the user name used when you signed on

From left to right the second line of the Status frame displays

- the current default path

  The default path consists of the node name, volume name (disk name), directory name, and (optionally) file prefix, which the Executive assumes you wish to use if they are not included in a file specification. (Node, volume, directory, and file prefix are described in the subsection "File Naming" in Section 4, "Basic Concepts.")

  The default path includes the file prefix. If no file prefix is displayed, the system assumes there is none.

- If the Executive is in a special state, such as Recording or Replaying, a message appears at the middle of this line indicating the state. This space is usually blank.
- The date and time to which the system clock is set.

EVENT FRAME

Messages from other programs or applications can appear in the Event frame in the double bar, which separates the Status frame from the Command frame. You will be aware of the Event frame only if a message appears.

Example:

```
_______________You have mail_________________
```

COMMAND FRAME

The area of the screen below the double bar makes up the Command frame. When no command is active the Executive shows a command prompt. When you give a command by typing the command name and pressing GO, the specific command form for that command is displayed. The screen diagram, Figure 3-1, shows the command prompt, with the command "Copy" and the command form for the Copy command.

As you execute commands, the sequence of commands and command forms is retained on the screen. New commands are added at the end of the current sequence. When the entire screen is full, information scrolls off the top of the command frame.

SCREEN CHARACTERISTICS

Screen characteristics, such as screen length, width, color etc. can be changed with the Screen Setup command. The options available vary with workstation model.

Automatic Screen Turn Off

The automatic screen turn-off feature causes the screen to go blank if no interaction with the workstation takes place for more than a specified period of time. The application is still present and the screen can be redisplayed by pressing any key. SHIFT is recommended, since it does not change anything on the screen.

You can set automatic screen turn off with the Screen Setup command. The time delay is variable.

KEYBOARD

Figure 3-2 shows the keyboard. Some of the keys have special functions when you are using the Executive.

TYPE-AHEAD

The operating system supports a feature called type-ahead. A buffer is used to store any keystrokes that the operating system cannot respond to immediately. The operating system looks in the buffer for keystrokes first before reading keystrokes directly from the keyboard. Thus, if your screen is blank and you are waiting for an Executive command prompt, you can type "Copy" and press RETURN. When the Executive screen is displayed, the operating system will read your keystrokes from the type-ahead buffer and will fill in the command field, recognize the RETURN and display the Copy command form. You can clear the type-ahead buffer if you want to by pressing ACTION-DELETE.

SPECIAL KEYS

ACTION-GO — If Context Manager is running, press ACTION and GO together while you are in the Executive, to display the Context Manager screen over the Executive screen.

ACTION-DELETE — Press ACTION and DELETE at the same time to clear your type-ahead buffer.

ACTION-FINISH — Press ACTION and FINISH at the same time to terminate the present application or operation. Your User Configuration file specifies what environment you return to after an ACTION-FINISH.

ACTION-OVERTYPE — Press ACTION and OVERTYPE at the same time to turn off the screen. Press any key to turn the screen on again.

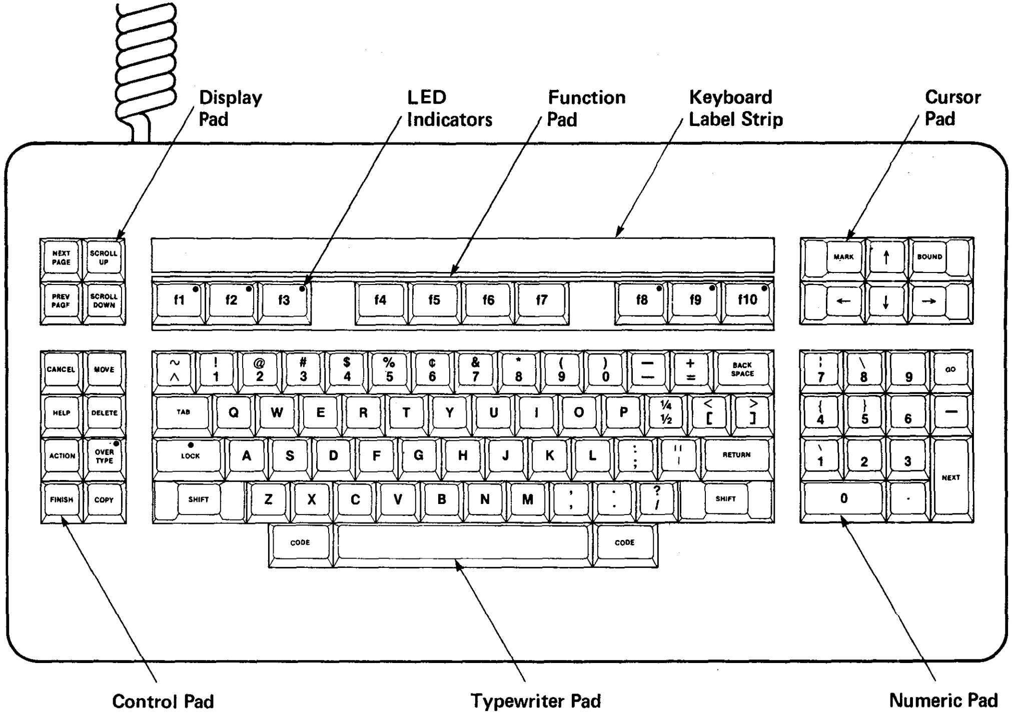


Figure 3-2. Keyboard.

Arrow keys

The up arrow, down arrow, right arrow, and left arrow keys move the cursor on the screen in the direction they indicate.

BACKSPACE

Press BACKSPACE to delete the previously typed character.

CANCEL

Press CANCEL to stop filling in a form and return to a new command form without executing the original command. The Executive may also prompt you to use CANCEL during a specific operation.

CODE-GO

CODE and GO together are used to activiate the Debugger. If you have the Debugger installed on your workstation, you can activate the Debugger by filling out the Executive command form and pressing CODE-GO rather than GO to execute the command.

DELETE

Press DELETE to delete all the characters you typed into the current field.

f1

f1 activates the Executive Redo feature. When you press f1 the Executive returns the last command form executed, with completed command fields ready for you to execute by pressing GO. You can also modify the parameters before you execute the form.

GO

Press GO to signal that you have completed a command form and wish the Executive to execute that command.

NEXT

Press NEXT to move the cursor to the next field of the command form, just as you would use RETURN.

NEXT PAGE

The Executive does not scroll information off the screen until you have had a chance to read it. If a command is about to scroll new data off the top of the frame during its execution, the message

> Press NEXT PAGE or SCROLL UP to continue

appears at the bottom of the screen, and scrolling is suspended. (New data here means any data displayed since your most recent keystroke.) Press NEXT PAGE to display the next page, SCROLL UP to display the next line, or CANCEL to skip the rest of the display.

A good example of this is the Type command, which displays the contents of a text file on the screen. It shows one screenful of text at a time, and then waits for you to press NEXT PAGE before going on to the next screenful of text. (This pause-between-pages facility can be disabled using the Screen Setup command described in Section 6, "Commands.")

RETURN

Press RETURN to end a line of input and move the cursor to the next field.

When you are giving a command, press RETURN to have the Executive display the specified command form.

SCROLL UP

The Executive does not scroll information off the screen until you have had a chance to read it. If a command is about to scroll new data off the top of the frame during its execution, the message:

> Press NEXT PAGE or SCROLL UP to continue

appears at the bottom of the screen, and scrolling is suspended. (New data here means any data displayed since your most recent keystroke.) Press NEXT PAGE to display the next page, SCROLL UP to display the next line, or CANCEL to skip the rest of the display.

SHIFT-GO

If you use Shift and GO together to execute a command, rather than using GO, then the Executive does not store the command in the memory buffer used for the Redo feature. Thus, if you execute command x using GO, then execute command y using SHIFT-GO. When you press f1 to invoke the Redo feature, the command form returned is the one for command x.

TAB

Press TAB to move the cursor to the next field of the command form, just as you would use RETURN.

EXECUTING A COMMAND

OVERVIEW

Executing an Executive command is a simple process of filling in a form.

1. Respond to the Executive command prompt by typing the command name and pressing RETURN.

2. Type the requested information into the command form the Executive displays.

3. Press GO to signal that you want the command executed.

EXECUTIVE COMMAND LIST

Commands are invoked by the Executive. Some of these are implemented within the Executive program itself. Others automatically invoke application Run files for you.

Your User Configuration file references a Command file, which determines the list of commands you can use. Your system administrator can add commands to this Command file using the New Command command. Commands can also be removed using the Remove Command command. For more information, see the subsection "Adding a New Command" in Section 5, "Advanced Concepts."

A complete list of commands you can use appears when you press HELP. The Help feature is described below.

Section 6, "Commands," describes all the Executive commands in detail.

EXECUTIVE HELP FEATURE

The HELP feature provides

- a list of commands available to you
- a description of each command
- a description of the specific command you just gave

A list of all the available commands appears on your screen when you press HELP. If you press HELP a second time, a description of the function of each command is also displayed.

If you have partially filled in the name of an Executive command when you press HELP, only those command names that match the partially completed command name are displayed. To display the description for a command at this point, you must type the command name or part of the command name and press HELP again.

GIVING A COMMAND

Figure 4-1 shows the way the screen appears when you execute the Executive command Copy.

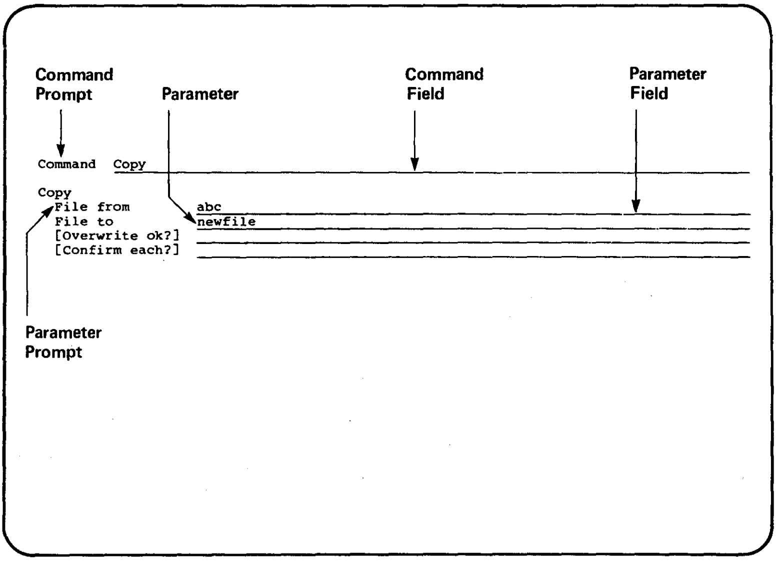
For most commands the Executive requires the name of the command and additional information about the task before it can execute the command. The items of information you provide are called parameters.

The Executive uses a command prompt to ask you for the command name. To tell the Executive which command you want to use, you type the command name into the command field. In Figure 4-1 the command Copy has been typed into the command field.

If the command you want requires no parameters, it can be executed after you complete the command field by pressing GO.

If parameters are required, you press RETURN after entering the command name in the command field. (You can optionally press the down arrow key, TAB, or NEXT, rather than RETURN.)

The Executive then displays the command form for the command name you gave. The command form consists of the command name and one or more parameter prompts. To complete the form you type the parameters requested into the appropriate parameter fields.



```
Command  Copy

Copy
  File from         abc
  File to           newfile
  [Overwrite ok?]
  [Confirm each?]
```

Figure 4-1. An Executive Command.

In Figure 4-1, File from, File to, [Overwrite ok?], and [Confirm each?] are all parameter prompts and the lines are parameter fields.

When the Executive finishes executing a command, it returns a new command prompt to the screen.

To cancel a command before executing, press CANCEL. The Executive displays the command prompt, so that you can give another command.

ENTERING THE PARAMETERS TO THE COMMAND

Parameters

A parameter is a string of characters representing an arbitrary value, such as a file name, a device name, a password, a "yes", or a "no".

To enter parameters you type them into the parameter field in response to the parameter prompt.

When you specify more than one item in a parameter field, each of them is referred to as a subparameter. You must include a space between each subparameter.

Characters Allowed

Parameters can include all characters. The space ( ), back slash (\), and single quote (') must be treated specially to be entered as a part of a parameter or subparameter, because you also use them to signal special functions to the Executive.

Spaces can be included in a parameter if each subparameter containing space(s) is enclosed in single quotes.

Example:

```
'The word filename'
```

A single quote or a backslash can be included in a quoted simple parameter by typing the character twice.

Example:

```
'The word''s a filename'
```

Use of the backslash (\) is discussed in the subsection "Executing a Command: Advanced Techniques" in Section 5, "Advanced Concepts."

Types of Parameter Fields

Optional Parameter Fields. If the parameter prompt is enclosed in square brackets ([]), completing the parameter field is optional. [Overwrite ok?] and [Confirm each?] are both optional parameter fields of the Copy command.

Each optional parameter field has a default value, which the Executive uses if you leave the field blank. You can find default values for each command in Section 6, "Commands."

If all the parameter fields to a command are optional, you can execute the command by typing the command name into the command field and pressing GO.

Yes/No Parameter Fields. Some parameter prompts in a command form include a question mark (?). You complete the parameter fields for these prompts by typing "yes" or "no". You can abbreviate these to "y" and "n". The default value for these fields is always no. [Overwrite ok?] and [Confirm each?] are both examples of fields that require a "yes" or "no" answer.

Examples

To add the contents of a file named Thisfile to a file named Thatfile, you enter parameters to the Append command as follows:

```
Append
  File list from    Thisfile
  File to           Thatfile
  [Confirm each?]   y
```

Execute the command by pressing GO.

To add each of a list of files to a file named Thatfile, you enter parameters as follows:

```
Append
  File list from    Firstfile Secondfile
  File to           Thatfile
  [Confirm each?]   y
```

Execute the command by pressing GO.

In this example, Firstfile and Secondfile are subparameters. Thatfile is a parameter.

GIVING A COMMAND TO INVOKE AN APPLICATION SYSTEM

Commands that invoke application systems, such as the Word Processor or Multiplan, are given in the same way that you give other Executive commands.

Some application systems can be invoked without completing parameter fields, since they can prompt you for this information themselves after they are invoked. In these cases, you can type the command name into the command field and press GO.

ABBREVIATING A COMMAND NAME

You can abbreviate the command name when you complete the command field. **The Executive needs only as many characters as are necessary to uniquely identify the command.** If more than one word appears in the command name, each word can be abbreviated.

Example:

For the command Asynchronous Terminal Emulator, you can complete the command field as shown below:

Command a t e

If you attempt to abbreviate a command but the Executive cannot match the abbreviation to a unique command, it displays the group of commands that match, and you can try again.

USING THE EXECUTIVE REDO FEATURE

The Redo feature allows you to recall to the screen the last Executive command form you used. To use Redo, press f1. The form is redisplayed with the parameter fields completed as you last typed them. You can change the contents of the parameter fields or execute the command exactly as before.

If you use SHIFT and GO together to execute a command, rather than using GO, then the Executive does not store the command in the memory buffer used for the Redo feature. Thus, if you execute command x using GO, then execute command y using SHIFT-GO, when you press f1 to invoke the Redo feature the command form returned is the one for command x.

OTHER ADVANCED FEATURES

You can use several advanced techniques to make command execution quicker and easier. They include substituting the contents of a file for a list of subparameters, entering keyboard characters such as GO, and using the Submit facility, which repeats a series of keystrokes stored in a Submit file. These are described in Section 5, "Advanced Concepts."

FILE NAMING

FULL FILE SPECIFICATION

To refer to a file when using the Executive you use a file specification. The file specification is the logical name for the file and indicates its location. File specifications can be given in their full form or abbreviated.

A full file specification has the form

```
{Nodename}[Volname]<Dirname>filename^password
```

where

- {Node} is the name of the node where the file is located.

  A node is a location in a network of workstations. It can be a master or a standalone workstation. There can be more than one cluster located at each node.

  A cluster workstation is located at the same node as its master and other workstations in the cluster. However, because communications cannot be initiated down the line from the master to the cluster workstation, a file on a cluster workstation cannot be accessed from a different node by including {Node} in the file specification.

- [Volname] is the name of the volume where the file is located.

  A volume is a disk drive that has been formatted and initialized to store files. It may be a hard disk, floppy or magnetic tape drive.

- <Dirname> is the name of the directory to which the file is assigned.

  A directory is a collection of related files on one volume. A volume is divided into one or more directories. A directory is actually a directory or index that stores the location of each file assigned to it.

- filename is the logical name you assign to the file.

A file is a set of related records treated as a unit. A file can contain a series of executable instructions, a series of data items, a document, etc.

- ^password is a valid password for the file. The password should also be valid for the command being executed. ^password is optional.

The distinction between upper- and lowercase is not significant for matching file specifications, but is preserved as entered and can be altered with the Rename command.

Node, Volume, and Directory Names

Node, volume, and directory names can all have 12 characters. It is suggested that the characters be alphanumeric, with the exception of the period (.) and the hyphen (-).

File Names

A file name can have up to 5Ø characters. It is recommended that characters be alphanumeric only with the exception of the period(.), the hyphen (-), and the greater-than sign (>).

Conventions. You can assign a file any name you wish so long as it is less than 5Ø characters long.

The system does not require any file name format except a valid file specification,as described above. However, you may wish to follow the conventions used with the Convergent application systems. Some of these are as follows:

- A period (.) is used to indicate that the suffix which follows describes the function of the file.

  Examples:

      filename.run, filename.sub, filename.tmp,
      filename.User

- An exclamation point (!) is used before a volume name to indicate that the volume is on the master workstation.

  Example:

  ```
  [!dØ]<sys>sys.cmds
  ```

- A dollar sign ($) is used as a prefix for directory names by the operating system.

  Example:

  ```
  <$ØØØ>
  ```

File Prefixes as a Part of a Filename. The Path and Set File Prefix commands allow you to set a prefix to be added by default to all file names. You can set any prefix you wish.

Prefixes are useful if you want to see lists of your files grouped conceptually. When you use the Files command to retrieve a list of files, you can use a wild card character to limit the list to files that include your prefix. The wild card characters are discussed below.

Example: You can create a conceptual subdirectory with the name "Personnel" by creating files with the names Personnel>reviews and Personnel>resumes.

ABBREVIATED FILE SPECIFICATIONS

You can abbreviate the full file specification, by including only as many parts of the file specification as the Executive needs to identify the file.

You can abbreviate by using one of the following formats:

- [volname]<dirname>filename

- <dirname>filename

- filename

The SignOn program sets default values for volume and directory when it reads your User Configuration file. You can change these values temporarily with the Path command. The Path command also allows you to specify a default node. Default node, volume, and directory are displayed for you in the Executive Status frame. These values can be referred to as the logged-in volume and directory.

If you leave node name, volume name, or directory name unspecified, the Executive assumes that you want to use the default values.

Example: If the Status frame shows you logged into [Sys]<sys> and you give the Copy command as follows, then the the full file specification assigned to the new file is [Sys]<sys>Fortran.example.

```
Copy
  File from          [fØ]<ct>Fortran.example________
  File to            Fortran.example________________
  [Overwrite ok?]    _______________________________
  [Confirm each?]    _______________________________
```

WILD CARD CHARACTERS

You can include a special character called a wild card character in a file specification . This tells the Executive that it should search for file specifications that match all characters given in the file specification **except** the wild card character(s). Where a wild card character appears, all matches are considered valid.

The Executive recognizes two wild card characters:

- o The asterisk (*), which specifies all matches of zero or more characters.

- o The question mark (?), which specifies all matches of exactly one character.

Wild card characters can be used whenever a command form prompts you for a file specification or file list. Wild card characters can be used as all or part of directory names and file names. You can use up to eight wild card characters in a file specification.

Thus, you can use the wild card character to specify a limited group of files, as in the specification [volname]<dirname>*.personnel, or to specify all files within a volume, as in the specification [volname]<*>*.

The simplest way to learn about wild card characters is to test them using the Files command.

Example: Figure 4-2 shows a list of files in a hypothetical directory and a list of possible specifications including a wild card. If the Executive includes the file in a list of files matching the specification given, the match is indicated with an "x".

Files in Directory	File Specifications with Wild Cards			
	*hisfile	?hisfile	Th*f??e	*file
Thisfile	x	x	x	x
thatfile			x	x
hisfiles				
hisfile	x			x
Thhisfile	x		x	x

Figure 4-2. Wild Card Examples.

For the Copy and Rename commands, you must be careful when using the wild card characters, because a wild card character used in one field must be used in the same way in the corresponding field.

CAUTION

Be especially careful when using the wild card characters with the Delete command. You can easily delete more than you intended to. You can try your wild card character specification with the Files command first, to check that only the files you intend to specify are indeed included. It is also a good idea to use the [Confirm Each?] option with the Delete command when using a wild card file specification.

Expansion of Wild Card Characters

If you enter a wild card character as part of a file specification in a parameter field, the Executive expands the file specification (or finds all possible matches) as soon as you move the cursor out of that field. The list of files that match the file specification you gave is shown in the parameter field. If the list is too long for the field, then an ellipsis (...) is displayed between the beginning and the end of the list.

Exception: When an Executive command is created with the New Command command, an option is available to suppress expansion of wild card characters until GO is pressed. Wild cards are not expanded then for these commands until you press GO .

FILE MANAGEMENT TECHNIQUES

The Executive offers several commands you can use for file management.

CREATING AND DELETING FILES

For many commands, if you specify a file that does not exist, it is created for you. Example: When you use the Editor command, it prompts for a file specification. If the file does not already exist, the Editor creates one for you.

You can also use the Create File command to create a file whose contents are undefined.

Files can be deleted separately or as a part of the files list with the Delete command. With the Remove Directory command, you can delete a directory and all the files in it.

LISTING YOUR FILES

You can use the Files command to obtain a list of all the files you have in any directory. The default files listing is an unsorted list of the files on the default directory.

To get a shorter, more logical list, use a file specification including a wild card character.

Example: If your directory contains the files exec.commands, status.August, exec.overview, and memo.binder, you can get a list of all the "exec" files by using the file specification exec.*

If you choose the [Details?] option, the listing displayed includes length of the file, number of sectors, last date modified, and protection level.

Any files list can also be saved in a file by using the [Print File] option of the Files command. The file can later be printed or viewed on the screen.

COPYING FILES

The Copy command copies the contents of one file to another file. If you enter "yes" in the

[Overwrite ok?] field, the contents of the second file will be overwritten.

LCopy provides a more sophisticated copying facility that copies a file from one volume or directory to another more easily than Copy, when you do not want to change the filename. It also optionally compares the source and destination file to verify the copy operation.

To merge one file with another file without writing over the contents of the second file you can use the Append command.

An entire floppy disk can be copied to another using the Floppy Copy command.

COPYING FILES FOR BACKUP

The Backup and Selective Backup commands both provide a specialized type of copy operation that copies files to a multivolume data set known as an archive file. Restore copies files from this archive file back to a working volume. Regular backup operations should be performed to prevent loss of important data. For information on how to back up your files, see the Backup Volume and Selective Backup commands in Section 6, "Commands."

DISPLAYING THE CONTENTS OF A FILE

To display the contents of a file without modifying it, you can use the Type command or the Dump command. Type displays the file one screenful at a time. The Dump command displays files in hexadecimal and ASCII simultaneously.

MODIFYING A FILE

The Editor or Word Processor are usually used to modify a file. The Maintain File command can also be used to modify files by reclaiming space occupied by deleted or malformed records in data files.

PRINTING A FILE

To print the contents of a file you can use

- direct printing
- spooled printing
- the facilities offered by each application system

Direct Printing

Direct printing prints a file directly from the workstation to a printer attached to that workstation. The printer must be unoccupied before each direct printing job can be started.

Before you try to use the direct printing process, check to see if your workstation is directly attached to a printer.

To print a file using the direct printing process you can use the Copy command or the Format command. If you use the Format command you can paginate the document, specify margins, etc. The Copy command transfers the contents of the file with no additional formatting. See Section 6, "Commands," for details.

Spooled Printing

Spooled printing uses a queue system to transmit the contents of the file to the printer. With spooled printing, workstations in a cluster can print to a locally attached printer or to printers attached to other workstations in the cluster system. When a spooler has been installed at a workstation for a printer, that printer is no longer available for direct printing.

To print a file using the spooled printing process you use the Print command, the Format command, or the Spooler Status command. See Section 6, "Commands," for details.

Printing a File Created Within an Application System

Some application systems, such as the Word Processor, have their own printing facilities. These can be configured for direct or spooled printing, but have their own commands for printing files. If you create a file within an application system, you should use the facilities it provides for printing.

For more information on the printing processes and installing the spooler, see the subsection "Printer Management" in Section 5, "Advanced Concepts."

FILE PROTECTION

PROTECTION SYSTEM

The CTOS operating system offers a file oriented security system.

Passwords control access to a specific device, volume, directory, or file.

Protection levels assigned to each file define the type of access allowed.

Using passwords and protection levels together you can define a file security system to meet your specific needs.

PASSWORDS

Four types of password protection are available:

- volume
- directory
- file
- device

Passwords can have a maximum of 12 characters. It is recommended that passwords be only alphanumeric with the exception of the period (.) and the hyphen (-).

If a password has been assigned, a valid password is required for certain commands, such as Backup Volume, IVolume, Create Directory, and the User File Editor. If you forget to supply the password or supply an incorrect one you will get the error message "Access denied (Error 219)".

Volume Passwords

The volume password allows you access to all files on the volume. If there is no volume password, there is no protection on the system. The volume password overrides directory or file passwords. In addition, if a volume password exists, it is required for creating directories or for opening the volume as a device.

For example, signing on with the volume password, or entering it with a Path command, gives you access to all files on that volume, whether they are password protected or not, without additional directory or file passwords.

NOTE

You must have a volume password for directory or file passwords to take effect.

Volume passwords are initially assigned when the volume is created using the IVolume command, and can be changed using the Change Volume Name command.

Directory Passwords

The directory password is used to restrict creating or renaming of files within a directory. If there is a directory password, you must specify it or the volume password to create or rename any files within the directory. A directory or volume password is required to remove a directory. A directory password can also be used to access a file, unless a protection level that ignores directory passwords has been assigned to the file. Protection levels are discussed below.

A directory password is established with the Create Directory command.

Directory passwords can be changed or removed using the Set Directory Protection command.

The default protection level set for files created within a directory has no relation to the directory password. Protection levels are discussed below.

File Passwords

The file password may allow access to a specific file, depending on protection level. Files do not have passwords when they are created.

To add a password to a previously unprotected file, or to change a file password, use the Set Protection command. This command also allows you to protect a list of files with the same password.

File passwords are most often used to allow certain files in a directory to be read, without allowing access to the other files.

Device Passwords

The device password is used for operations that work directly with the disk, such as the IVolume or Backup Volume commands. These passwords are assigned when the operating system is built. Unless you have a customized operating system, default passwords assigned with standard software apply. For the hard disk the password is the same as the device name. Example: DØ or D1. For floppy disks, the default is no password.

PROTECTION LEVELS

File protection level controls which types of passwords are required, if any, to gain read or modify access to a specific file.

Protection level is assigned to files only. A directory can have a default protection level; however, this is used to assign a protection level to each file at the time that the file is created.

The protection level is a number used by the operating system. Protection level numbers are based on bit numbers and are not hierarchical. For more information on how the operating system uses protection levels, see the CTOS Operating System Manual.

Nine protection levels are available. Table 4-1 shows name, number, and type of access allowed for each protection level.

The "Unprotected" level allows Read and Modify access without a password.

Three levels allow Read access without a password, but require a password for Modify access. They are

- Modify Protected
- Modify Password
- Nondirectory Modify Password

Five levels require a password for both Read and Modify Access. They are

- Access Protected
- Access Password
- Read Password
- Nondirectory Password
- Nondirectory Access Password

File passwords are ignored by three of the protection levels; directory passwords are ignored by three levels.

A default file protection level can be assigned to a directory. This does not in any way affect the passwords and protection of the directory, but is used only as a default level for files created within the directory. If a directory has a password and is assigned the lowest level of protection (15 "unprotected") it is **not** truly unprotected, since a directory or volume password is still required to create or rename files within that directory. When created, files within that directory are assigned a protection level of 15 "unprotected," but the protection level can be changed with the Set Protection command.

Table 4-1. Protection Levels.

Protection	Level	Password Required	
Level	Number	To Read	To Modify
Unprotected	15	None	None
Modify Protected	5	None	Volume or directory
Nondirectory Modify Password	23	None	Volume or file
Modify Password	7	None	Volume, directory, or file
Access Protected	Ø	Volume or directory	Volume or directory
Read Password	1	Volume, directory, or file	Volume or directory
Nondirectory Access Password	19	Volume, directory, or file	Volume or file
Access Password	3	Volume, directory, or file	Volume, directory, or file
Nondirectory Password	51	Volume or file	Volume or file

Assigning and Changing Protection Levels

Each file is assigned a protection level by default at creation. The default protection level is originally set with the Create Directory command.

After a directory has been created, the default protection level can be changed with the Set Directory Protection command.

You can change the protection level for a specific file or file list within your directory with the Set Protection command.

USING A PASSWORD FOR ACCESS

If no volume password was assigned to the volume when it was initialized, you can sign on to the system without supplying a password and have full access to all files.

If a volume password is assigned, a volume, directory, or file password can be entered when you sign on. Access is restricted accordingly.

The Path command can also be used to give a password. Thus, if you have signed on with a directory password and wish to access files in a different directory, you can supply the necessary password with the Path command.

Some commands include parameter prompts for a password.

You can also enter a password as a part of a device, volume, directory, or filename. If a parameter or subparameter includes a caret (^) the characters between the caret and the end of the name are considered to be the password. Example: filename^password

PROTECTING YOUR SYSTEM

More information on protecting your file system is included in the subsection "System Security" in Section 5, "Advanced Concepts."

EXECUTING A COMMAND: ADVANCED TECHNIQUES

ENTERING SPECIAL KEYBOARD CHARACTERS

You may want to enter a keyboard character such as GO when completing a parameter field.

The back slash (\) is used to indicate to the Executive that the value of the next keyboard character typed is to be included in the parameter.

Thus, to include characters such as GO, you can press back slash (\) followed by GO. The Executive will not show the backslash on the screen when it is typed.

To include the backslash (\) as a character in a simple parameter you must type it twice.

Example:

```
'Type a backslash \\'.
```

SUBSTITUTING THE CONTENTS OF A FILE FOR A LIST OF SUBPARAMETERS

The contents of a file can be used to replace a parameter in a parameter field.

The at-sign (@) is used to signal to the Executive that the parameter following is a file specification. When you move the cursor to the next parameter field, the Executive reads the file specified and uses the contents as subparameters.

Files used in this way are called "at-files."

At-files are useful when you need to give more subparameters than the command form allows space for or when you use a certain parameter or list of subparameters frequently.

Example

To use an at-file to list a series of files to be appended to another file, you create a file in the

Editor containing a list of all the files you want to append. If you call that file Filename you fill out the form like this:

```
Append
  File from          @Filename
  File to            Longfile
  [Overwrite ok?]    y
  [Confirm each?]
```

You execute the command by pressing GO.

SPECIFYING A DEVICE AS A PARAMETER

Devices can be specified as parameters to commands. To specify a device, you enclose the device name in square brackets ([]), in the same way that you specify a volume.

Devices you can specify include

o	the screen	[KBD]
o	the keyboard	[VID]
o	printers	[PTR]A [PTR]B [LPT]
o	communications channels	[COMM]A [COMM]B
o	the spooler	[SPL]
o	disks	[DØ] [FØ]
o	the null device	[NUL]

Example

In the example below, the Copy command is used to transfer characters from the keyboard to a printer using direct printing.

```
Copy
  File from          [KBD]
  File to            [LPT]
  [Overwrite ok?]
  [Confirm each?]
```

Specifying a device this way uses byte streams and the Sequential Access Method. For more information on byte streams, see the CTOS Operating System Manual.

USING A SUBMIT FILE

A Submit file is used to automatically repeat a sequence of keystrokes. Submit files are most often employed to repeat a sequence of Executive commands.

Whenever you frequently use the same set of commands, you may want to use a Submit file. Examples are daily backup and restore, or compiling, linking, and running a program.

Submit files are created with the Record command or in the Editor. To invoke a Submit file, you use the Submit command.

When you use the Submit command to invoke a Submit file, the Executive reads characters directly from the file, rather than from the keyboard. Submit files are automatically closed when all characters have been read.

Within a Submit file you can

- include comment text

- submit other Submit files and include recursive calls to the same file

- use variables to represent parameter values so that they can be changed at the time the Submit file is executed

- use conditional variables to represent parameter values

- interrupt the sequence of commands within the Submit file to allow characters to be entered directly from the keyboard

A Submit file is said to be expanded when a variable used to signify a parameter is replaced with an actual parameter value assigned at the time of execution.

To use the Submit facility you use the Record, Stop Record, and Submit commands.

CREATING A SUBMIT FILE USING THE RECORD COMMAND

The Record command allows you to execute a sequence of Executive commands and, at the same time, to record those commands in a Submit file, which can be executed again at a later time. This is the easiest way to create a Submit file.

A Submit file created with the Record command can be modified later with the Editor, if you wish.

Note: A Submit file created using a 132-column screen may not work with an 8Ø-column screen, since the lines of the command forms may be too long.

To create a Submit file follow this procedure:

1. To give the Record command type Record and press RETURN.

   The Executive will display:

   ```
   Record
     File to record on  ____________________________
   ```

2. Type the name you would like to assign to the Submit file in the parameter field. The name must be a valid file specification. If you use a file name already assigned, the old file will be overwritten.

       Press GO.

   Note that the Status frame displays the message "RECORDING." This means that the Executive is storing each keystroke of the command sequence in the Submit file.

3. Execute the series of Executive commands you wish to record, just as you would if you were not in record mode.

4. Give the Stop Record command when you have completed the entire sequence of commands.

5. To execute the Submit file, give the Submit command by typing Submit into the command field and press RETURN.

   The Executive will display:

```
Submit
  File                  ______________________________
  [Parameters]          ______________________________
  [Force Expansion?]    ______________________________
  [Show expansion?]     ______________________________
```

Type the file name you specified for the Submit file in step 2 above in the File field.

Press GO

The Executive executes the sequence of commands exactly as before.

Note that the Status frame displays the message "REPLAYING".

CREATING A SUBMIT FILE USING THE EDITOR

You can create a Submit file in the Editor by typing the sequence of commands and parameters into the file as you would type them in the Executive.

Special characters such as GO must also be entered. This is done in the Editor using the Insert command (CODE-I) to insert a hexadecimal keyboard code. See the Editor Manual for more information on how to use the Editor. The keyboard codes are listed in Appendix D, "Keyboard Codes."

Any Submit file originally created with the Record command can be modified with the Editor.

To use the Editor to create a Submit file follow this procedure:

1. Give the Editor command by typing Editor and pressing RETURN.

   The Executive will display:

   ```
   Editor
     File name      ______________________________________
     [Your name]    ______________________________________
   ```

2. Type the name you wish to assign to the Submit file in the File name field.

   Press GO

The Editor creates the file and displays a blank screen for you to type into.

3. Type the first command name you wish to have the Submit file invoke. Press RETURN. The Editor inserts the RETURN symbol. (You can see the RETURN symbol if you press CODE-v.)

4. Type in the parameters necessary for the command you have chosen in exactly the same way that you would enter them in the parameter fields. A new line is necessary only when the Executive Command form would show a new line.

5. When all the necessary parameters have been included, use the Editor Insert command to insert the code 1B, the hexadecimal code for the GO character.

   Repeat steps 3-5 until you have entered all the commands you wish to give in your Submit file.

6. FINISH the Edit session.

7. Try your Submit file by invoking the Submit command and specifying the name of the file you just created. Note that execution errors will be harder to find if you create a file this way rather than with the Record command.

USING AN ESCAPE SEQUENCE

An <u>escape sequence</u> is a sequence of characters that invoke special functions when read from a Submit file. These functions include

- comment
- nonconditional expansion
- conditional expansion
- character input directly from the keyboard

The percent sign (%) is usually used to signal to Submit that the sequence of characters following is an escape sequence.

The cent sign (¢) is used to indicate that the sequence of characters following is an escape sequence read directly from the keyboard.

INCLUDING COMMENTS IN A SUBMIT FILE

To include comments in a Submit file use the percent-sign (%) to signify an escape sequence follows and enclose the text in single quotes or double quotes.

Example:

```
    %'This is a comment' or %"This is a comment"
```

You can omit the closing single quote if you replace it with a RETURN to include the RETURN as a part of the comment. This allows you to have a comment that is more than one line long.

Example:

```
    %'This is also a comment RETURN
```

You can omit the closing double quote if you replace it with a RETURN, also, but in this case the RETURN is not taken as a part of the comment.

Example:

```
    %"This is also a comment RETURN
```

NONCONDITIONAL EXPANSION OF A SUBMIT FILE

To include a variable parameter in a Submit file, use

o the percent sign (%) to signify an escape sequence follows, followed by

o the numbers Ø - 9 as variables

When you invoke the file, use the Parameters field of the Submit command to enter the value you wish to use for the variable Ø, followed by a space, followed by the value for 1, followed by a space, and so on.

Example:

```
Submit
  [Parameters]  file1 file2 file3
```

You can also use the greater-than character (>) before the variable number to indicate that all variables greater than the one named should be expanded to their assigned value. With this option, you can indicate a file list within the Submit file without listing all the variables separately. This allows you to use more than nine variables.

Example: See Appendix B, "Examples," for an example of how to create and execute a Submit file with nonconditional expansion.

The following escape sequences are expanded to set values:

- %[is expanded to the default volume name.
- %< is expanded to the default directory name.
- %^ is expanded to the default password.
- %U is expanded to the User Name.

CONDITIONAL EXPANSION OF A SUBMIT FILE

Escape sequences can include conditional variables.

Sequences of conditional expansions can be nested within each other, but must be balanced and cannot overlap. Nested conditional expansions are expanded from the inside out.

To include conditional variables in your Submit file type

1. the percent sign (%) to signify an escape sequence, followed by

2. a question mark (?) to indicate a condition where the parameter exists, or a pound-sign (#) to indicate a condition where the parameter does not exist, followed by

3. the value you wish to substitute for the variable if the condition is true, enclosed in a set of delimiting characters of your choice

You can use the greater than character (>) in the same way that it is used with nonconditional expansion.

Examples:

%?3/one/ is expanded to 1 if parameter 3 exists and to null otherwise

%#1/%2/ is expanded to the value of parameter 2 if parameter 1 does not exist and to null otherwise

%?>4/five/ is expanded to 5 if any parameter greater than 4 exists and to null otherwise

INTERRUPTING SUBMIT TO ENTER CHARACTERS DIRECTLY FROM THE KEYBOARD

A special escape sequence that begins with the cent sign (¢) is used to signify that you wish characters to be read directly from the keyboard.

To create a Submit file that will accept input at designated points directly from the keyboard, type

1. a cent-sign (¢) to indicate that an escape sequence will follow, followed by

2. a number one (1) to signify that you wish to read characters directly from the keyboard, followed by

3. the character you wish to use to signify that characters should be read from the Submit file again. An example might be RETURN or GO.

Example:

¢1 and the RETURN key

This example shows an escape sequence that could be used to allow the operator to enter a particular parameter or a command.

You can terminate a currently running Submit file and return character input to the keyboard, by using the cent sign (¢) to signify that an escape sequence will follow, plus the number two (2).

Example:

```
    ¢2
```

NOTE

If you use a ¢1 escape sequence in a file that invokes one or more other Submit files, any occurrences of ¢1 after the second invocation of Submit will be interpreted as ¢2 escape sequences and your Submit file will be closed.

CREATING A SUBMIT COMMAND TO INVOKE A SUBMIT FILE

You can use the New Command command to create a Submit command to invoke a Submit file. For a discussion of this procedure see the subsection "Adding a New Command," below.

INVOKING USER PROGRAMS

CREATING A RUN FILE

User programs are originally created in the Editor or Word Processor. The source module is either assembled, compiled, or interpreted into an object module. The Linker is then used to link one or more object modules into a task image that can be executed by the operating system. The resulting file is called a Run file.

See the Editor Manual, Word Processor Manual, and Linker/Librarian Manual for more information.

INVOKING A RUN FILE

To invoke a Run file, you can use the Run command.

The Run command builds a Variable Length Parameter Block (VLPB) from which the Run file can read the parameters in the same way that it would if you had created a command with the New Command command. (For a discussion of the VLPB, see the section on parameter management in the CTOS Operating System Manual.)

If the Debugger is installed on your workstation, it can be used to help debug Run files. The Debugger is used with the Run command. To activate the Debugger, complete the command form and press CODE-GO instead of GO. For more information on the Debugger, see the Debugger Manual.

You can also use the New Command command to create a command to invoke the Run file. See the subsection "Adding a New Command," below.

ADDING A NEW COMMAND

The Executive recognizes commands defined in a file called a Command file. You add information to the Command file using the New Command command, and remove it using the Remove Command command.

The Command file is located on the system directory (<Sys>). The default file provided is named [Sys]<Sys>Sys.cmds.

Commands can be implemented within the Executive program itself or by a separate Run file. Submit files can be implemented as commands using a special procedure and a special Run file called [Sys]<sys>Subcmd.run.

CUSTOMIZED COMMAND FILES

If you wish, the system can be configured so that different users can access different Executive Command files. This is done by creating a customized Command file and specifying its name in the User Configuration file. See the subsection "User Configuration," below, for more information on the User Configuration file.

A customized Command file can be created by copying [Sys]<sys>Sys.cmds to a new file and adding commands using the New Command command, or removing commands using the Remove command command.

ADDING A COMMAND TO INVOKE A RUN FILE

Most often, commands are added by users to invoke Run files. To do this the New Command command is used to specify the command name, Run file name, parameter prompts (called field names), HELP description, and case value, and the name of the Command file to which the command is to be added.

Case value is a two-character string used to identify the command if the Run file could be called by more than one command.

Example: Two hypothetical commands, X and Y could invoke the same Run file, XandY.Run. During the creation of the X command, the case value 'ØØ' is used. During the creation of the Y command the case value 'Ø1' is used. The XandY Run file when invoked can query the case value used in the command to determine which command invoked it. For more information on case value see the section on parameter management in the CTOS Operating System Manual.

When the parameter prompts are specified, you can also indicate whether wild card characters should be expanded when the cursor is removed from that parameter field. Expansion can be delayed until GO is pressed by adding an asterisk (*) to the parameter prompt name. The asterisk is not displayed on the resulting command form.

ADDING A COMMAND TO INVOKE A SUBMIT FILE

You may also wish to add a command to invoke a Submit file you use frequently, such as a cleanup and backup routine.

To do so you use the New Command command and specify the file [Sys]<sys>SubCmd.run as the Run file.

The name of the Submit file used for the command must be of the form [Sys]<Sys>Name.Sub, where "Name" is the command name, with any embedded spaces removed.

Parameters entered in the New Command form are indicated to Submit using escape sequences (i.e. %Ø, %1, %2 ...), as as described in the subsection "Using a Submit File," above. Parameters are placed in the escape sequences in the following order:

1. first subparameter of each parameter field in the form from top to bottom

2. remainder of the subparameters of each parameter field in the form from top to bottom

Example: See Appendix B, "Examples," for an example of a command to invoke a Submit file.

ADDING A COMMAND IMPLEMENTED WITHIN THE EXECUTIVE PROGRAM

The default Executive Command file already contains the command information for all the commands implemented within the Executive. The procedure below is described for your information and will rarely be used.

Commands implemented by the Executive program itself are added to the Command file in the same way that you add a command to implement a Run file, except that you do not specify a Run file name. For these commands you enter a character string made up of an exclamation point (!) and a number. Table 5-1 below shows the character string used for each command.

Example:

The command form used to add the Type command is

```
New Command
  Command name      Type
  Run file          !6
  Field names       'File list' '[Confirm each?]'
  Description       'Display one or more files on
                    the video display.'
  [Overwrite ok?]
```

Table 5-1. Commands Implemented Within the Executive.

Command Name	Run File Field
Append	!1
Copy	!2
Create Directory	!14
Create File	!5
Delete	!3
Path	!7
Record	!19
Remove Directory	!15
Rename	!4
Run	!18
Screen Setup	!17
Set File Prefix	!9
Set Protection	!1Ø
Stop Record	!2Ø
Type	!6

STATUS INQUIRY

System status can be monitored on several levels through

- the Executive Status frame
- status commands
- error logs

EXECUTIVE STATUS FRAME

The Executive Status frame is the most frequently used status information facility. From the Status frame you can verify which version of the operating system and which version of the Executive is currently running, the user name used at SignOn, the current default volume, directory, and file prefix, and the date to which the system clock is currently set.

STATUS COMMANDS

The Volume Status command is used to obtain information on a volume and the directories in that volume. Data displayed includes creation date, last modification date, number of free pages, and number of free file headers. The names of the directories on the disk and the current default protection level for each directory are also given.

If you have a multipartition operating system, user memory can be divided into partitions. Each application is executed in a separate partition. The Partition Status command is used to monitor partition status as well as to perform partition operations. The Partition Status command gives the number of current partitions, their size, and the name of the Run file currently executing in each partition. It also gives memory size and boundaries for each partition.

Cluster activity can be monitored with the Cluster Status command. An overview display shows active cluster lines and the user name given at SignOn for each line. A more detailed display also shows the number of lines and workstations configured, line utilization, errors, and, for each specific user, the number of request frames received since the user became active and the number of outstanding requests.

The Spooler Status command provides information on the status of the printer spooler queue and is also used for controlling some printer operations. Status information includes lists of files queued for each printer, messages from the printer, and whether the printer is printing, idle, offline, or paused.

ERROR LOG

Various types of system errors and events generate a message stored in a file called a Log file. The Plog command allows you to examine the contents of the Log file. Plog displays the entire contents of the file, or will show information on selected types of errors. This Log file should be inspected periodically to identify marginally functioning hardware or data base errors.

USER CONFIGURATION

A User Configuration file uniquely identifies each user to the system. This file is used by the SignOn program and by application systems such as the Word Processor. The User Configuration file entries define the application environment and other options such as default path, Executive Command file, and messages displayed at SignOn.

Each user is assigned a unique user name, which is associated with a specific User Configuration file. The user name can be any string of characters up to 3Ø characters in length. A user may have more than one user name assigned.

The User Configuration file is always located on the System directory ([Sys]<sys>). The file specification consists of the user name given at SignOn plus the suffix ".User".

Example:

```
    [Sys]<sys>Allen.User
```

A sample User Configuration file is shown in Figure 5-1.

User Configuration files can be created and modified using the User File Editor or the standard file Editor.

The User File Editor allows you to select from the available application environments and modify User Configuration file entry options. With the User File Editor, you can create a User Configuration file containing a series of default entries, or you can create a customized User Configuration file.

USER CONFIGURATION FILE FORMAT

Each entry in the User Configuration file defines one system characteristic to be used by various application programs, such as SignOn. The entries for SignOn and the Executive are described below. For information on entries required for other application systems, see the appropriate reference manual.

```
:SignOnTextFile:[sys]<sys>System.Txt
:SignOnChainFile:[sys]<sys>Exec.Run
:SignOnExitFile:[sys]<sys>SignOn.Run
:SignOnVolume:DØ
:SignOnDirectory:Allen
:SignOnFilePrefix:
:SignOnPassword:
:MailPrinters:[sys]<sys>Mail.printers
:MailCenterName:Hub
:MailUserName:Allen M.
:MailPassword:Password
```

Figure 5-1. Sample User Configuration File, [Sys]<Sys>Allen.User.

The format for entries in the User Configuration file is ":Keyword:Data".

Example:

```
:SignOnVolume:Sys
```

For an entry to be recognized, the colon (:) must appear in the first column of the line. Any other text in the file is ignored when the file is read.

If you use the User File Editor it formats the User Configuration file for you. The User File Editor also assigns default values for the User Configuration file entries.

User Configuration File Entries for SignOn

SignOn entries are

```
:SignOnVolume:
:SignOnDirectory:
:SignOnFilePrefix:
:SignOnPassword:
:SignOnChainFile:
:SignOnTextFile:
:ExecCmdFile:
:SignOnExitFile:
```

The SignOn entries are described in detail below. The entries for SignOnChainFile, SignOnTextFile, and ExecCmdFile are optional.

Default Path. All User Configuration files must contain entries for default volume name, directory name, file prefix, and password. These entries define the default path and are used by the operating system whenever a full file specification is not made. (See Section 4, "Basic Concepts," for an explanation of default paths.)

The data field of the file prefix and password entries may be left blank, but the keyword must be included. If the data field of the Password entry is left blank, then SignOn uses any password typed into the SignOn password form as the default password.

Example:

```
    :SignOnVolume:Sys
    :SignOnDirectory:Sys
    :SignOnFilePrefix:
    :SignOnPassword:
```

Exit File. A SignOnExitFile entry must appear and a file name must be specified. The file specified as SignOnExitFile is loaded by the operating system whenever a program terminates with an Exit or ErrorExit, or whenever ACTION-FINISH is done.

Example:

```
    :SignOnExitFile:[Sys]<sys>SignOn.Run
```

All invalid optional entries are ignored by the SignOn program.

Text File. The contents of the file specified in the SignOnTextFile entry are displayed on the screen by the SignOn program after you sign on, before it chains to the Chain file.

If you leave the data field blank by not specifying a file name, then text in the Configuration file is used for the message until the next colon (:) is encountered.

Example:

```
    :SignOnTextFile:System.Txt

    :SignOnTextFile:
    ###
    ###The text above is from System.Txt.  The
    ###text below is from Messages.Txt.
    ###
    ###

    :SignOnTextFile:Messages.Txt
```

If you include each entry shown in the example above, SignOn will display three messages.

If you use the User File Editor you cannot create more than one SignOnTextFile entry that specifies a file name. If you specify multiple SignOnTextFiles the User File Editor will choose the first one.

Chain File. The file named in the SignOnChainFile entry is the file that is loaded after you sign on to the system and SignOn has displayed any text files specified. SignOn invokes the exit file after you exit from the chain file.

If you do not specify a chain file the SignOn program loads the Exit File.

Example:

```
    :SignOnChainFile:[Sys]<Sys>WordProcessor.Run
```

In this example, SignOn loads the Word Processor immediately after you sign on.

The chain file can include parameters to be passed to it when it is invoked. Parameters are passed to the chain file in the same way they are passed in the Executive parameter interface. For this reason, the first line following the SignOnChainFile must contain the name of the command to be invoked by the Executive. The parameters passed to the file start on the next line and are terminated by a line starting with a colon (:).

Example:

```
    :SignOnChainFile:[Sys]<Sys>Submit.Run
    Submit
    Init.Sub
    arg0  arg1


    :
```

The last example has the same effect as filling out the Submit form from the Executive as follows:

```
Submit
  File list              Init.Sub
  [Parameters]           arg0 arg1
  [Force expansion?]     ____________________________
  [Show expansion?]      ____________________________
```

Since SignOn cannot expand wild card characters, you cannot include them here in parameters passed to the file.

Executive Command File. The Executive Command file may be different for each user. If this entry is missing or invalid, the Executive uses the file [Sys]<sys>Sys.Cmds.

Example:

```
    :ExecCmdFile:[Sys]<Sys>Allen.cmds:
```

This example specifies that the Executive uses Allen.cmds.

User Configuration File Entries for Application Systems

Entries can also be made in the User Configuration file specifying options for application systems, such as the Word Processor. Each application system may have its own default values to use for these options. The entries need only be specified if you do not want to use the defaults. See Release Notices or Reference Manuals for such applications for a description of necessary entries.

DEFAULT USER CONFIGURATION FILE

A default User Configuration file is provided with standard software. The name of the default file is [Sys]<Sys>.User. The SignOn program looks for the user name typed into the SignOn form plus the suffix .User. If no name is supplied, it will use the default file, [Sys]<Sys>.User, if it exists. [Sys]<Sys>.User, therefore, should be deleted after valid User Configuration files have been created for the current users on your system.

USER FILE EDITOR

The User File Editor can be used to create or modify User Configuration files. It allows you to select application environment options and to select entries to be included in the User Configuration file. The environment is defined by the combination of the SignOnExitFile entry and the SignOnChainFile entry. Environment choices are offered through one menu and entry options through another. For a description of how to use the User File Editor, see Section 6, "Commands."

The initial environment is the application loaded after SignOn. In some cases this may be your total environment. For example, you may go directly into the Word Processor after SignOn and return to SignOn immediately when you FINISH. In other cases, an additional application may be loaded when you exit from the first. As an example, the Word Processor may be loaded immediately and the the Executive can be loaded after you FINISH the Word Processor.

User Configuration files not created with the User File Editor can still be modified through it. If the User File Editor does not recognize a field in such a file, it leaves it out of the modified file created and gives you a message to that effect.

Files created by the User File Editor have an additional entry with :Environment: as the keyword, This precedes the SignOnExitFile and SignOnChainFile entries. The :Environment: entry is used by the User File Editor to identify the environment. If this entry is not included, the User File Editor will call the environment "unnamed".

The entry options define the User Configuration file entries and how those entries appear in the User File Editor menus.

User File Editor Template File

The menus displayed by the User File Editor are defined by a Template file. The Template file is set up by the system administrator to define the choices of user application environments and the User Configuration file entry options.

Most users can use the User File Editor without understanding the Template file.

The Template file includes default values for environments and entry options, so that the User File Editor can create files with a default format.

You can set up more than one template file, and can specify which one to use when you give the User File Editor command.

A default template file is provided with standard software. The default file is named [Sys]<Sys>Userfiletemplate.sys. If no template file is specified when you invoke the User File Editor, the command uses this default file.

Template File Format

Figure 5-2 shows the format of a typical template file.

Template file entries are grouped into two divisions, Environments and Options. The Environment division, which is required, describes the available environments. The Options division describes User Configuration file entry options, which may optionally be broken down into several categories.

The order in which the environment and category sections of the template file are given determines the order in which the menus are displayed.

You can modify the name of the Environments menu and the name of the Options menu by changing the data item in the Template file.

```
:Options:Modify Options
:Category:Executive
    :Field:ExecCmdFile

:Category:Mail
    :Field:MailUserName
        :Description:Mail user name
    :Field:MailCenterName
        :Description:Mail center name

:Category:SignOn
    :MustField:SignonVolume
        :Description:Volume
        :Default:sys
    :MustField:SignonDirectory
        :Description:Directory
        :Default:sys
    :MustField:SignonFilePrefix
        :Description:File prefix
    :MustField:SignonPassword
        :Description:Password
    :Field:SignonTextFile
        :Description:Text file
        :Default:[sys]<sys>signon.txt
        :Description:Command file

:Environments:Select Environment
    :Name:Executive
        :SignonExitFile:[sys]<sys>Exec.run
        :SignonChainFile:[sys]<sys>Exec.run
    :Name:Word Processing
        :Command:Word Processor
        :SignonExitFile:[sys]<sys>exec.run
```

Figure 5-2. Example of a Template File for the User File Editor.

If only one category appears, a single option entry form is displayed, rather than an Option menu showing only one option.

The format for template file entries is ":Keyword:Data". The entry is recognized if the colon (:) is the first character on the line, other than a space or a tab. Entries are hierarchical and must appear in the order shown in the example. All "data" entries can be changed, with the exception of field names, which are required to appear in the User Configuration file.

Options. Each category becomes a choice shown in the Options menu. If only one category is given, an Options menu is not displayed and an option entry form appears instead.

"Keyword" can be

o :Options:

  The value following "Options: appears in the Main menu as a choice and in the Options menu as a title.

  Example:

```
:Options:Modify Options
```

o :Category:

  Categories are the names you want to use for the group of entries listed below it. Each category is an option you can choose from the Option menu displayed by the User File Editor.

  Categories are displayed on the menu in the order they are listed in the Template file.

  When you use the User File Editor, after choosing a category you can edit the various fields described below it. Typically, categories represent application system choices, such as electronic mail.

  The order in which the following fields are given determines an option entry form for each category.

  - :Field:

    Fields are the actual entries added to the User Configuration file. When you use the User File Editor, the entries specified by ":Field:" can be edited after you choose an option from the Options menu.

    :Field: entries will be written out only if the "data" entry is nonblank.

- :FileField:

  When an entry is a required User Configuration file entry, and a file name for that entry must be specified, :FileField: must be included in the Template file, and a file name must be included to be used as a default.

- :MustField:

  If an entry is a required User Configuration file entry but does not require a file name, the field is specified in the Template file as a :MustField:, and the User File Editor will include the entry in all User Configuration files it creates, whether the data entry name is actually specified or not.

- :Description:

  Description is the text used to describe the entry on the User File Editor form. It is a friendly name for the field.

- :Default:

  Default is the value the User File Editor will use if no other value is specified for the entry when a new User Configuration file is created. These values will be included in the configuration file created unless they are changed through the User File Editor.

Environments. Environments determine the SignOnChainFile SignOnExit file. The Environment division is required.

"Keyword" can be

o :Environments:

  The value following :Environments: appears in the Main menu as a choice and in the Environment menu as a title.

- o :Name:

  Name is the name you choose for each type of environment available. The text will appear in the Environment menu.

  Following a name there are fields:

  - :Command:

    Command must be a command name to the Executive. The User File Editor will expand from this to produce the SignOnChainFile, SignOnExitFile, and lines of parameters if necessary. Therefore other fields are not necessary unless you want to override them.

    The :Command: field is used to specify the initial environment.

    Example:

    ```
    :Command:Word Processor
    ```

  - :SignOnExitFile:

    The file specified as SignOnExitFile is loaded by CTOS whenever a program terminates with an Exit or ErrorExit, or whenever ACTION-FINISH is done.

  - :SignOnChain File: This field is used only in the case where you do not have a :Command: specified. The file specified as SignOnChainFile is loaded by SignOn after any SignOnTextFile.

  - :Default:

    No value can be given here.

    If you include this entry, the User File Editor uses the environment in which :Default:" appears as the default environment for User Configuration files when no other is chosen. If you do not include ":Default:" the User File Editor takes the first field listed in the template file as the default. If you include ":Default:" more than once, the first one listed is the valid one.

SYSTEM PROTECTION

The protection system operates by restricting access to the system itself through means of user names and passwords used at SignOn and through use of passwords and protection levels assigned to specific files.

File system security can be established on different levels by

- o restricting access to the system
- o protecting your files from other users of the system
- o protecting files from access by sophisticated systems programmers

RESTRICTING ACCESS TO THE SYSTEM

System access can be restricted to authorized users. This is done with the SignOn interface, which requires that the user have the knowledge of a password and a user name to gain access. To achieve this, assign a password to the [Sys] volume. Each User Configuration file should have protection level 51 assigned and should have a password. (Protection levels are described in Section 4, "Basic Concepts.")

In order to gain access to the system the SignOn program requires read access to a user file. If you assign protection level 51 to the user file then either the User Configuration file password or the volume password is required.

PROTECTING YOUR FILES FROM OTHER USERS

The next class of protection protects your files from access by other users of the system. Here the protection is achieved by setting up a User Configuration file for each user with a password unique to that user. In addition, it entails creating a directory for each user with a password assigned to it. Each user knows his or her user file password and directory password (which could be the same), but not the volume or other directory passwords.

When the user's directory is created it should have a default protection level of either Ø (no access to files without volume or directory password), or 5 (no modification of files without directory or volume password). This default is the protection level that will be assigned to files that are created within the directory.

PROTECTING YOUR FILES FROM ACCESS BY SOPHISTICATED SYSTEMS PROGRAMMERS

The final class of protection is from sophisticated users of the system. Here the limiting factor is the ability to run a program at the workstation with the physical file system attached (the master of a cluster). In general, this means physical access to the workstation.

From a cluster workstation, the file system at the master is intended to be secure from users without knowledge of the volume or <Sys> directory passwords. The <Sys> directory password must be secure in order to guard against unauthorized users reading the operating system files contained in the <Sys> directory. These files must have protection level Ø and are [Sys]<Sys>Mfd.Sys and [Sys]<Sys>FileHeaders.Sys.

VOLUME MANAGEMENT

Four commands are used for volume management: IVolume, Backup Volume, Selective Backup, and Restore. They are designed to initialize, back up, and restore volumes and files. These commands are also used to verify the integrity of volume control structures and to eliminate volume fragmentation. (See the subsection "Volume Fragmentation" below). The volume commands are described in detail in Section 6, "Commands."

ARCHIVE FILE

Backup Volume, Selective Backup, and Restore use a multivolume data set generically referred to as an archive file. An archive file actually consists of several files located on one or more volumes.

The archive file consists of 1 to 99 floppy disks with the same volume name and consecutive numeric suffixes (.Ø1, .Ø2, etc.). For example, one archive file can consist of [Archive]<Sys>.Ø1, [Archive]<sys>.Ø2, etc.

An archive file can also be on a magnetic tape volume or another larger disk.

INITIALIZING A VOLUME

The IVolume (Initialize Volume) command prepares a floppy or Winchester disk for use as a Convergent volume. IVolume formats the media to accommodate 512-byte sectors, performs write/read tests to identify surface defects, writes volume control structures (such as the Volume Home Blocks, the Master File Directory, the Allocation Bit Map, and File Header Blocks) onto the media, and creates system files. IVolume must be performed on a disk before it can be used.

BACKING UP FILES

Either the Backup Volume or Selective Backup commands can be used to backup files.

Backup Volume Command

The Backup Volume command copies all the files from one volume to an archive file and then verifies the integrity of the volume control structures. Backup Volume can archive all files or archive only those files modified on or after a specified date or date and time (incremental backup).

Additionally, Backup Volume can verify the integrity of the volume control structures without backing up the files, recognize a formatted volume that is too scrambled to be automatically recognized by the operating system, and optionally write the log of the Backup Volume operation to a file or printer. (The log remains on the screen.)

Regular use of Backup Volume is critical for the following reasons:

- It creates an archive file of the contents of a volume as a protection against disk hardware failure, inadvertent file deletion, etc.

- It verifies the consistency of volume control structures.

  Backup volume performs extensive checks for consistency on all volume control structures. Irregularities such as Allocation Bit Map inconsistencies, File Header Blocks (FHBs) without directory entries, or vice versa, and invalid FHBs are reported. If Backup Volume reports irregularities in a volume, determine the cause, reinitialize the volume with IVolume, and restore the files from the archive file.

- It eliminates volume fragmentation (in conjunction with IVolume and Restore). See the subsection "Volume Fragmentation" below.

See Appendix B, "Examples," for an example showing the procedure for backing up a volume.

Selective Backup Command

The Selective Backup command copies individual files or directories to an archive file. As opposed to Backup Volume, Selective Backup copies only individual files or directories. Users need only read access to the files to be archived. With Backup Volume device-level access is required.

Additionally, Selective Backup can archive only those selected files modified on or after a specified date or date and time (incremental backup). Optionally, Selective Backup can write the log of the Selective Backup operation to a file or printer.

Restoring Files

The Restore command restores files onto a volume from an archive file created by either Backup Volume or Selective Backup. Restore can restore an entire volume (creating any needed directories on the destination volume), restore selected files to the same or different file specifications, restore a file's characteristics (creation date, protection level, etc.) as they existed at the time of backup, and optionally write the log of the Restore operation to a file or printer.

Restore operates in either volume or selective restore mode. In volume mode, an entire archive file is restored to a volume. In selective mode, individual files or directories are restored.

VOLUME FRAGMENTATION

A basic understanding of the volume control structures is useful in appreciating the importance of regular volume reinitialization (the use of Backup Volume, IVolume, and Restore) to avoid volume fragmentation. See the section on file management in the CTOS Operating System Manual for a more detailed discussion of volume control structures.

When a request is made to the CTOS file system to create or extend a file, the operating system attempts to satisfy it by allocating a single disk extent. (A disk extent is one or more contiguous disk sectors that compose all or part of a file.) If that is not possible, it allocates two or more smaller disk extents whose total size is sufficient to satisfy the request.

A single disk extent is large enough for the operating system to satisfy the request, if a volume was initialized or reinitialized recently with IVolume. However, if files were created and deleted many times since a volume was initialized, the disk extents available for allocation may be scattered, making it impossible to satisfy the request by allocating a single disk extent. In this case the available storage of the volume is said to be fragmented.

If a volume becomes fragmented, performance is degraded in several ways:

- It takes longer to create or extend a file because the operating system must access more sectors of the Allocation Bit Map to allocate sufficient disk extents to satisfy the request.

- It takes longer to process a file sequentially because disk sectors that are logically consecutive are not physically consecutive.

- It limits the number of files that can be open concurrently. This is because each open file requires allocation of a File Area Block (FAB) in memory for each disk extent of the file.

  In order for the CTOS operating system to accommodate any maximum number of concurrently open files, a sufficiently large number of File Area Blocks must be specified during the system build. Increasing the number of File Area Blocks, however, reduces the amount of memory available to the application system.

See Appendix B, "Examples," for an example showing the procedure for reinitializing and restoring a Winchester disk from an archive file to reduce volume fragmentation.

CLUSTER MANAGEMENT

Workstations can be clustered to share the facilities of a printer, peripheral mass storage devices on the master, and communications devices.

A cluster is configured to have one master workstation, which polls the other workstations on the cluster. All communications pass between the master workstation and a cluster workstation; communication cannot occur directly between cluster workstations.

A cluster workstation can have its own disk and tape drives and can therefore stand alone, or it can share these resources with the master workstation.

The Disable Cluster command causes the operating system in the master workstation to stop communicating with the cluster workstations and closes all files that are open for the cluster workstations, thus disabling the cluster operations. This is most often used when installing software or making modifications to the master workstation. The Resume Cluster command reenables cluster operations.

The Cluster Status command can be used from any cluster workstation to monitor cluster activity. It tells you who is signed on to the cluster as well as giving you information about the communications lines.

PARTITION MANAGEMENT

The partition commands support the simultaneous execution of several application systems, each in its own partition. Partition commands can be used only if you have a multipartition operating system.

Partition commands allow you to

- create, load, and remove secondary partitions and obtain partition status information

- terminate all tasks in a secondary partition and leave the partition vacant

For more information on partition management, see the CTOS Operating System Manual.

COMMANDS AVAILABLE

The partition commands are

- Create Partition, which creates a vacant secondary partition

- Partition Status, which contains four subcommands that do the following:

  - display status information for one or all partitions (Display Partition Status)

  - load and activate a run file in the specified vacant partition (Load Primary Task)

  - terminate all tasks in the specified partition but do not load and activate the partition's exit run file (Vacate Partition)

  - remove the specified vacant partition (Remove Partition)

PRINTER MANAGEMENT

A printer can be configured for direct or spooled printing. You can invoke either process from the workstation, through the Executive, using the appropriate command. When printing is invoked through an application system that includes its own printer commands, such as the Word Processor, a direct or a spooled printing process is still used. Printing a file is discussed in the subsection "File Management" in Section 4, "Basic Concepts."

Direct or spooled printing both require a printer interface through which the commands can communicate with the printer. Printer interfaces can be parallel (Centronics-compatible) and serial (RS-232-C-compatible).

A printer interface requires that the printer be attached to a workstation and a printer configuration file be created to be used to define the printer characteristics. (For a discussion of configuration files, see the Create Configuration File command in Section 6, "Commands.")

Spooled printing, in addition, requires a Spooler Configuration file and the installation of two system services, the Queue Manager and the Spooler.

When printing is invoked through an application system, such as the Word Processor, a special configuration file, called Sys.Printers, is also used.

For more information on spooled printing, see the CTOS Operating System Manual.

DIRECT PRINTING

Direct printing can only occur from the workstation on which the file resides to a printer directly attached to that workstation. Before a file can be printed with the direct printing process, the local printer must be idle. During the direct printing process, no other activity can occur on the workstation.

The commands used to invoke direct printing are Copy and Format. Application systems that have their own printing commands can also use direct printing. For these systems you use the commands they provide.

Direct printing can be accessed through the Sequential Access Method using printer byte streams, a sequence of 8-bit bytes, to transfer text from memory to the printer interface. (See the section on the Sequential Access Method in the CTOS Operating System Manual.)

Example:

```
Copy
  File from          myfile____________________________
  File to            [Lpt]_____________________________
  [Overwrite ok?]    __________________________________
  [Confirm each?]    __________________________________
```

Printer Interfaces for Direct Printing

The names used for each of the printer interfaces for direct printing are

- [Lpt] direct printing to the parallel printer interface
- [Ptr]A direct printing to the channel A serial printer interface
- [Ptr]B direct printing to the channel B serial printer interface

Configuring a Printer for Direct Printing

To set up a printer for direct printing you must

- attach the workstation to the printer and verify that they are communicating
- create a Printer Configuration file or modify the default Printer Configuration file provided with standard software, so that the printer commands can communicate properly with the printer

Attaching the Printer to the Workstation. For direct printing in a cluster environment, the printer is attached to an individual workstation. For information on physically attaching the printer to the workstation see the Installation Guide and the Processor Manual for your workstation. After the printer is attached to the workstation, you can verify that it is communicating with the workstation by using byte streams or, for a serial printer, the Asynchronous Terminal Emulator. (See the Asynchronous Terminal Emulator Manual).

Printer Configuration Files. A configuration file, defining the printer characteristics, must be established for **each new printer.** Some characteristics defined include number of characters printed per minute, baud rate, and line control mode. All the commands that are used for printing need to use this file to communicate correctly with the printer.

Default Printer Configuration Files. Three default printer configuration files are installed with the standard software. These can be used for a new printer and can be customized if necessary. The names for these files are

- [Sys]<sys>PtrAConfig.Sys for [PTR]A
- [Sys]<sys>PtrBConfig.Sys for [PTR]B
- [Sys]<sys>LptConfig.Sys for [LPT]

For more information on these files, see the Create Configuration File command in Section 6, "Commands."

Overriding the Default Configuration File. The default configuration file specification associated with each printer can be overridden when the printer byte stream is opened with the OpenByteStream operation. (See the CTOS Operating System Manual.) To override the default file, append &[volname]<dirname>filename to the device specification, where [volname]<dirname>filename specify the configuration file to be used.

Example:

```
Copy
  File from          [DØ]<documents>mymemo
  File to            [Ptr]B&<Sys><sys>newconfig.sys
  [Overwrite ok?]    ______________________________
  [Confirm each?]    ______________________________
```

SPOOLED PRINTING

Spooled printing transfers text to a disk file for temporary storage and queues a request for the text to be transferred to the first available printer interface under its control.

Spooled printing allows workstations to share printers. It also allows workstation activity to continue while the file is being printed. While direct printing requires that you wait until a printer is available, spooled printing allows you to issue a request for printing at any time, then proceed to other activities.

When a spooler has been installed at a workstation for a printer, that printer channel will, generally, no longer be available for direct printing. However, you can release a printer channel for direct printing or other use through the Spooler Status subcommand Free Printer Channel.

A cluster can be configured several different ways for Spooled Printing:

- One printer can be shared by all workstations and can be attached to one workstation, either a cluster or a master.
- Two printers attached to the same workstation can be shared by all workstations in two ways:
  - They can use the same queue, if you do not want to differentiate between printers.
  - They can use different queues.
- Printers can be be attached to more than one workstation and
  - can use the same queue
  - can use different queues

Requirements for Spooled Printing

There are five requirements for spooled printing:

- The printer must be attached to a workstation and you must verify that they are communicating as described above for direct printing.

- A Printer Configuration file must exist that defines the printer characteristics.

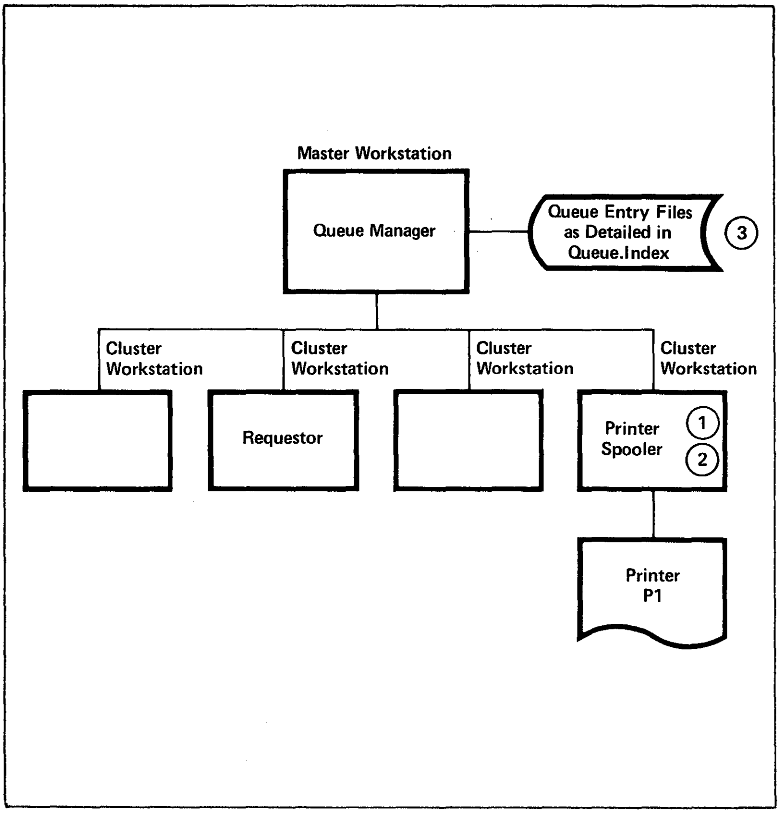
- A Spooler Configuration file must exist for each installation of the spooler to define the printer and printer channels used for each workstation.

- A list of files that the Queue Manager is to manage must be provided in the file [Sys]<Sys>Queue.Index at the master workstation. A default Queue.Index is provided with standard software and can be modified for your specific installation.

- Two system services must be installed, the Queue Manager and the spooler, using the Install Queue Manager and Install Spooler commands, respectively.

For information on attaching a printer and setting up a Printer Configuration file, see the subsection "Direct Printing," above.

Figures 5-3 and 5-4 show examples of two possible spooler configurations. Note the location of the Queue Manager, Spooler Configuration file(s), and Printer Configuration file(s) for each.



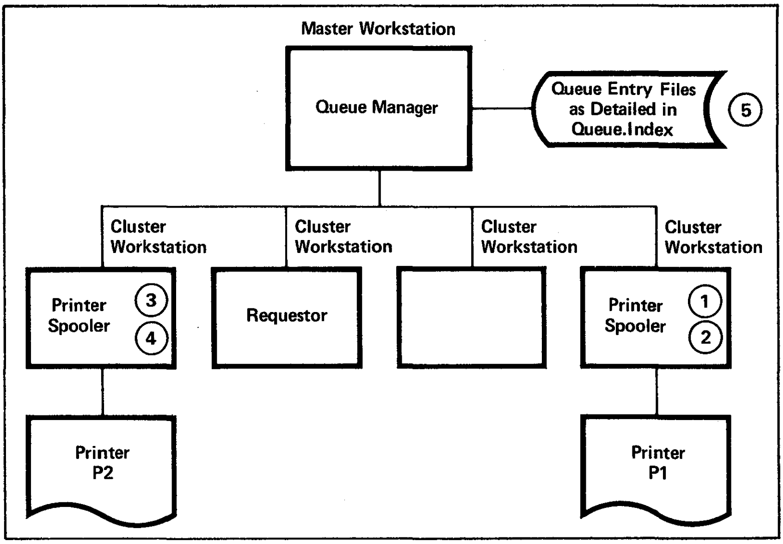
Note the corresponding entries for printer P1 and queue SPLA in Queue.Index and the Spooler Configuration File.

(1) Printer Configuration File
    [Sys]<Sys>PTRBConfig.Sys

(2) Spooler Configuration File
    [Sys]<Sys> SPLCnfg.Sys
        B/P1/SPLA/[Sys]<Sys>PTRBConfig.Sys/65/y

(3) Queue Index File
    [Sys]<Sys>Queue.Index
        SPLA/[SYS]<SYS> SPLA.QUEUE/1/1
        P1CONTROL/[SYS]<SYS>P1CONTROL.QUEUE/1/1
        SPOOLERSTATUS/[SYS] <SYS>SPOOLERSTATUS.QUEUE/1/1

Figure 5-3. Example of a Spooler Configuration Showing One Printer and One Queue.



Note the corresponding entries for printer P1 and queue SPLA in Queue.Index and the Spooler Configuration File for the first printer. The Queue.Index entries for P2 and SPLB similarly correspond to entries in the Spooler Configuration File for the second printer.

① Printer Configuration File for First Printer

```
    [Sys]<Sys>PTRBConfig.Sys
```

② Spooler Configuration File for First Printer

```
    [Sys]<Sys>SPLACnfg.Sys
      B/P1/SPLA/[Sys]<Sys>PTRBConfig.Sys/65/y
```

③ Spooler Configuration File for Second Printer

```
    [Sys]<Sys>SPLBCnfg.Sys
      B/P1/SPLB/[Sys]<Sys>PTRBConfig.Sys/65/y
```

④ Printer Configuration File for Second Printer

```
    [Sys]<Sys>PTRBConfig.Sys
```

⑤ Queue index File

```
    [Sys]<Sys>Queue.Index
      SPLA/[SYS]<SYS>SPLA.QUEUE/1/1
      SPLB/[SYS]<SYS>SPLB.QUEUE/1/1
      P1CONTROL/[SYS]<SYS>P1CONTROL.QUEUE/1/1
      P2CONTROL/[SYS]<SYS>P2CONTROL.QUEUE/1/1
      SPOOLERSTATUS/[SYS]<SYS>SPOOLERSTATUS.QUEUE/1/1
```

To make both printers share the same queue, SPLA, change the contents of ③ SPLBConfig.Sys to

```
      B/P1/SPLA/[SYS]<SYS>PTRBCONFIG.SYS/65/Y
```

and remove the second entry in Queue.Index ⑤.

Figure 5-4. Example of a Spooler Configuration Showing Two Printers and Two Queues.

The Spooler

The spooler services print requests and can simultaneously control the operation of several printers attached to a single workstation.

The spooler is a dynamically installed system service and is not resident when the operating system is bootstrapped. Each spooler consists of a set of processes that manages spooled printing of disk files. One process is assigned for each printer under control of the spooler.

If printers are to be used by all other workstations in the cluster system for spooled printing, the spooler is installed at each workstation to which a printer is attached. Therefore, in a spooled printing configuration, more than one spooler may be installed.

The spooler is installed with the Install Spooler command and must be installed after the Queue Manager. The spooler may be installed on the master workstation.

A Spooler Configuration file defines the printers associated with that workstation and defines the printer channels used. This file is used during the installation of the spooler. A default Spooler Configuration file is provided with standard software.

There is always one Spooler Configuration file per spooler. Although the spooler service resides in the memory of the workstation with the printer attached, the associated Spooler Configuration file may reside either on the master or on the cluster workstation disk.

The Queue Manager

The Queue Manager acts as a switch between print requests from a workstation and the spooler installed on the same or another workstation. When the spooler has a printer available, it obtains the next entry from the Queue Manager and prints the file.

The Queue Manager must be installed at the master workstation. Only one Queue Manager is installed in a spooled printing configuration.

The Queue Manager references a file called [Sys]<Sys>Queue.Index, which is on the Master and defines the queues used to control spooled printing. Entries in Queue.Index must exist for each queue required. Placing an entry in Queue.Index is all that is required to create the queue.

Three types of queues are defined in Queue.Index:

- scheduling

- control

- spooler status

Scheduling Queues. When a user Submits a request with the Format, Print, or Spooler Status command, the Queue Manager places the queue entry (the file specification for the file to be printed and parameters controlling the printing) in a scheduling queue entry file.

Each scheduling queue is defined by an entry in Queue.Index. There must be at least one scheduling queue for each class of printer. Thus, if two printers have the same characteristics, they may use the same scheduling queue. Multiple scheduling queues can be maintained.

Because the scheduling queue is disk-based, the printer spooler can resume printing automatically when reinstalled following an operating system reload.

A scheduling queue is associated with each class of printer and contains queue entries generated when the Format or Print command is executed. Queue entries are also generated when you used the Print subcommand of the Spooler Status command.

The association between a scheduling queue and its printer is defined in the Spooler Configuration file.

When giving the Format, Print, or Spooler Status command, you specify the name of the scheduling queue. When the command is executed, a queue entry is automatically generated by the printer spooler and sent to the Queue Manager. The Queue Manager places the queue entry in the appropriate scheduling queue entry file. When the printer spooler's printer becomes idle, it obtains a queue entry for the printer from the Queue Manager.

In addition to scheduling queues, the printer spooler requires control and status queues that are used for printer control and status inforation purposes.

Control Queues. A control queue is required for each printer in the system. Control queues store the printer control requests required by the spooler for processing. When you execute the Spooler Status command using its printer control subcommands, entries are automatically placed in the associated control queue entry file. The spooler queries the control queue entry file periodically to check for printer control entries.

The name of the control queue is the name of the printer appended by the string "Control." For example, if printers Joe and Tom share the same scheduling queue named Diablo, the printer spooler requires the queues JoeControl and TomControl.

Spooler Status Queue. A single, system-wide status queue named SpoolerStatus stores information about each spooled printer in the cluster configuration. Each spooler periodically updates the entry for each printer under its control and for special events. The status information contained in the SpoolerStatus queue is accessed when the user executes the Spooler Status command.

For more information on the Queue Manager, see the section on Queue Management in the CTOS Operating System Manual.

For information on accessing each queue from an application and on the format of each queue see Appendix C, "Programmer's Notes on the Spooler."

Names of Scheduling Queues and Printers

A scheduling queue name or a printer name is used in various subcommands of the Spooler Status command.

The system administrator names each printer and each scheduling queue to be used in the system. The association between the printer name and the scheduling queue name is defined by the Spooler Configuration file. When a printer becomes idle, its spooler obtains the next queue entry from the scheduling queue associated with that printer.

If several printers share the same scheduling queue, a file is printed on whichever of the printers becomes available first.

The names of scheduling queues and printers must be unique, except when a single printer is associated with a scheduling queue. In the latter case, the name of the scheduling queue and the printer can be the same.

Specifying Banner Page/Notice File

The spooler continuously transfers text from disk files to printers in the sequence of queue entries for that scheduling queue. The continuous printing results in a single stack of paper behind each printer, which consists of text from many disk files. To provide a visual indication of the beginning of each file, the spooler can print a banner page before printing the file.

The banner page is visually distinctive and identifies the file being printed. The banner page can also contain the text of a notice file.

You specify whether the banner page is to be printed through a parameter in the Spooler Configuration file or in the spooler commands.

The notice file is a convenient way of conveying operational information, such as the version of the software currently in use, to a later reader of the printed output. This is done using the file [Sys]<Sys> Spooler.Notice notice, an ordinary text file that can be created and modified by the Editor.

Files Required to Configure Spooled Printing

Queue.Index. [Sys]<Sys>Queue.Index is installed with standard software. You may modify the file or create a new one using the Editor.

Queue.Index stores

- The name of a system-wide status queue named SpoolerStatus, which stores information about each spooled printer in the cluster configuration. The Spooler Status queue is updated regularly by the spooler.

- The name of each scheduling queue used by the system.

- The name of the control queue associated with each printer. The Control queue is used to store printer control requests invoked through the Spooler Status command.

The format for Queue.Index is configured as follows:

- Only one Spooler Status queue name entry is needed.

- Each class of printers must have one scheduling queue entry.

- Each printer must have an associated control queue name entry.

- Each queue entry must start on a new line and must have the format

  ```
  QUEUENAME/FILESPEC/ENTRYSIZE/QUEUETYPE
  ```

  where

  QUEUENAME is the name of the print queue.

  FILESPEC is the name of the file specification for the queue named in QUEUENAME. These files must be located at the master' workstation.

  ENTRYSIZE is number of sectors per entry. The spooler and batch both use an entry size of 1.

  QUEUETYPE is 1 for spooler queue. For Batch queues see Section 7, "Batch Processing."

Examples:

Scheduling queue entry for print queue SPLB

```
    SPLB/[SYS]<SYS>SPLB.QUEUE/1/1
```

Control queue entry for printer named DIABLO

```
   DIABLOCONTROL/[SYS]<SYS>DIABLOCONTROL.QUEUE/1/1
```

SpoolerStatus queue entry

```
   SPOOLERSTATUS/[SYS]<SYS>SPOOLERSTATUS.QUEUE/1/1
```

Spooler Configuration File. Default Name: [Sys]<sys>Splcnfg.Sys. One Spooler Configuration file is required for each spooler you install. A spooler may be installed at any workstation where an associated printer is attached. The configuration file may be on the master or cluster workstation.

When the spooler is installed, the Spooler Configuration file must contain an entry for each printer channel to be controlled by the spooler (even if the printer is not configured at this time). The number of entries tells the spooler how much memory space to allocate for printer data structures.

Additional information required for each printer can be specified in one of two ways:

- o include it in the Spooler Configuration file when the printer spooler is installed

- o use the Spooler Status command or the ConfigureSpooler operation, which is described in the section on spooler management in the <u>CTOS Operating System Manual</u>.

The following additional information is required for each printer:

- o the name of the printer

- o the name of the scheduling queue

- o the Printer Configuration file specification

- o the priority of the printer control process (see the subsection "Control Queues" above)

- o whether a banner page is printed between files

The spooler configuration file is an ordinary text file that can be created and modified using the Editor or Word Processor. A default Spooler

Configuration file is provided when standard software is installed.

Each printer channel to be controlled by the printer spooler requires an entry in the Spooler Cnfiguration file.

The format for the Spooler Configuration file is

PRINTER CHANNEL/PRINTER NAME/QUEUE NAME/PRINTER CONFIGURATION FILE/PRIORITY/BANNER (Y/N)

- PrinterChannel is a single-character code that specifies the printer channel to which the printer is connected:

  - Ø is the parallel channel
  - A is channel A
  - B is channel B

  If the entry consists of the printer channel only, data structures will be allocated for future use of the specified channel, but the printer is not allocated for use by the spooler.

- Printername is the name of the printer connected to the printer channel. If a printer name is specified, the spooler allocates buffers and initializes the channel for its own use only.

- Queuename is the name of the scheduling queue in which entries for printing requests are stored until the printer is available.

- PrinterConfigurationFile is the file used to specify the printer configuration file that describes the device parameters.

- Priority is the priority (1Ø-254, with 1Ø the highest) of the printer spooler process for the printer. A priority greater than 128 (i.e., lower than the default priority of the interactive application system) ensures that the printer spooler does not impact the interactive application system.

- Banner is a "Y" or "N" (the default). "N" means that the banner page is not printed at the beginning of each file. "Y" means that the banner page is printed. (See the "Specifying Banner Page/Notice File" subsection above.)

Example:

```
B/QUME/[SPLQ]/[SYS]<SYS>SPLBCONFIG.SYS/128/Y
```

Printer Configuration File

One printer configuration file is required for each printer. A default configuration file is provided for each printer type with standard software. A Printer Configuration file can be created or edited using the Create Configuration File command. The Create Configuration File command will create the Printer Configuration file in the correct format.

SPOOLED PRINTING OF PASSWORD PROTECTED FILES

The Format, Print, and Spooler Status commands expand a file specification but do not append the logged-in (default) password before sending the queue entry to the Queue Manager. To permit the spooler to read a read-protected file, you must either specify security mode or append an in-line password to the file specification when you give the Print or Spooler Status command. If file security is a concern, security mode should be used.

If security mode is specified, the printer spooler pauses before printing the file and waits for a password to be entered. The printer spooler remains paused until you either

- use the Spooler Status command to enter the password, and execute the command from the workstation to which the printer is connected, or

- cancel the print request with the Cancel subcommand of the Spooler Status command

If the password is entered, the spooler completes the printing.

PRINTING MODES

Any of three printing modes can be specified in the Format, Print, or Spooler Status command: normal, image, or binary.

Normal mode prints the banner, converts tabs into spaces, converts end-of-line characters into device-dependent codes, and recognizes the escape sequences for manual intervention.

Image mode prints the banner and recognizes the escape sequences, but performs no code conversion.

Normally, characters such as tab and end-of-line are converted to the form expected by the particular kind of printer as specified by the Printer Configuration file. In some cases, it is necessary to override this conversion because sophisticated application systems sometimes know more about the specific printer than is encoded in the Printer Configuration file. For example, the Word Processor knows how to print proportionally spaced text on certain kinds of daisy-wheel printers. Image mode is provided to suppress all character conversions (except recognition of printer spooler escape sequences).

Image mode can be specified to spooler (or printer) byte streams through the SetImageMode byte stream operation. For compatibility between spooled and direct printing, the SetImageMode operation should be used before the first WriteBsRecord or WriteByte byte stream operation.

Binary mode does not print the banner, send any extra code not in the file to the printer, or recognize the escape sequences.

MANUAL INTERVENTION IN SPOOLED PRINTING

A printer under control of the spooler can require manual intervention for either intentional or unintentional reasons. Unintentional reasons include out-of-paper, offline, and paper-jam conditions. Intentional reasons include forms change, print wheel change, and generic printer pause.

Whenever a printer requires manual intervention, the Spooler Status command can be run to determine the cause. In a cluster configuration, the Spooler Status command can be run from any workstation.

After determining and correcting the need for manual intervention, printing can be restarted using the Restart Printer subcommand of the Spooler Status command.

Generic printer pause is initiated by the Spooler Status command. Forms change and print wheel change are initiated by printer spooler escape sequences. (See Appendix C, "Programmer's Notes on the Spooler.")

BOOTSTRAPPING

The Bootstrap command uses the operating system to load a program into memory, replaces the operating system with that program, and then transfers control of the workstation to that program. The program can be a diagnostic or another operating system.

Bootstrap can also be invoked by an operating system during initialization. In this case, there are no parameters from the Executive. (See below.)

INVOKING BOOTSTRAP DURING SYSTEM INITIALIZATION

To invoke Bootstrap during initialization, an operating system must specify [Sys]<Sys>Bootstrap.Run as the "InitFileSpec" at system build. In this case

o if the workstation was last bootstrapped from its local Winchester disk, Bootstrap loads

  [Sys]<Sys>Bootstrap>SysImage.Sys, or

o if it was last bootstrapped from a master workstation , Bootstrap loads

  [Sys]<Sys>WSnnn>Bootstrap>SysImage.Sys

  where nnn is the type of workstation.

If Bootstrap is used to install an operating system, the operating system must be appropriate for the workstation in its current configuration. For example, a standalone operating system must not be installed with the Bootstrap command if the current operating system is for a cluster configuration and is bootstrapped from the master workstation.

Swapping operating systems expect to find their nonresident portion in the file [Sys]<Sys>SysImage.Sys. To set up a two-stage bootstrap of a swapping operating system, the large swapping operating system must be copied into both SysImage.Sys and <Bootstrap>-SysImage.Sys. Then SysImage.Sys must be overwritten with a smaller first-stage operating system. (Its total size must be smaller than the resident portion of the swapping operating system.)

6 COMMANDS

INTRODUCTION

This section discusses each command in detail and shows the command form used to give the command. The commands are listed in alphabetic order.

Before using this section, you should read the subsection "Executing a Command" in Section 4, "Basic Concepts." "Executing a Command" describes in detail how to complete the parameter fields of a command form and discusses the conventions used to indicate whether a parameter field is optional and whether it takes a yes/no parameter.

APPEND

Introduction

The Append command adds each of the files you specify in the File list from field to the file you specify in the File to field. The first character of each file appended immediately follows the last character of the preceding file, so no artificial gaps are created.

Append may not work with files created within certain application systems. For example, word processing files cannot be used with the Append command.

The Append command is implemented internally by the Executive using the value !1.

Command Form

Append
File list from ____________________
File to ____________________
[Confirm each?] ____________________

Parameter Fields

File list from

Enter a file specification, file list, or an at-file name. These file(s) are appended to the file named in the File to field.

File to

Enter the file specification for the file to which you wish the file(s) named in the File list from field to be appended.

The file specified will be created if it did not previously exist.

[Confirm each?] Default: No

If you enter "yes", then you are prompted to confirm appending each file by pressing GO, to deny it by pressing CANCEL, or to stop appending files by pressing FINISH.

If you enter "no" or leave the field blank, then you are not prompted for confirmation.

BACKUP VOLUME

Introduction

The Backup Volume command copies files from one volume to an archive file and verifies the integrity of the volume control structures.

Backup Volume can

- Archive all files.

- Archive only those files modified on or after a specified date, or date and time (incremental backup).

- Verify the integrity of the volume control structures without backing up the files.

- Recognize a formatted volume that is too scrambled to be automatically recognized by the operating system.

- Optionally write the log of the Backup Volume operation to a file or printer. (The log always appears on the screen.)

This command is implemented using the Run file BackupVolume.Run.

Command Form

Backup Volume
Volume or device name ______________
[Volume or device password] ______________
[Incremental from
(e.g., Mon Jun 1, 1981 8:ØØ pm] ______________
[Suppress backup?] ______________
[Suppress verification?] ______________
[Archive file] ______________
[Delete existing archive file?] ______________
[Log file] ______________
[Display structures?] ______________

Parameter Fields

Volume or device name

Enter the name of the volume or device you wish to back up.

[Volume or device password]   Default: No password

Enter the password (if one exists) of the volume or device you wish to back up.

The volume password must be supplied if the volume is recognizable by the CTOS operating system. Otherwise, the device password must be specified.

[Incremental from (e.g., Mon Jun 1, 1981 8:00 pm)]

Default: Backup all files

Enter the date from which files are to be backed up. Only files modified on or after the specified date are backed up.

A time can also be optionally specified after the year.

If no date is specified, all files are backed up. If no time is specified, all files from 12:00 am (midnight) of the specified day are backed up.

[Suppress backup?]   Default: No

If you enter "yes", the integrity of the volume control structures is verified without a backup being performed. The results appear on the screen and, if specified, in the log file.

[Suppress verification?]   Default: No

If you enter "yes", Backup Volume performs only the backup pass, and suppresses the verification pass.

The verification pass should be suppressed only when a full backup is being done, and only if the volume will be reinitialized with the IVolume command immediately following the backup. Otherwise, the integrity of the volume cannot be ascertained.

In any case, any errors encountered during the backup pass are listed on the screen.

[Archive file]   Default: [Archive]<Sys>.nn

Enter the name of the archive file to be created.

If no file is specified, Backup Volume creates one named [Archive]<Sys>.nn.

Since the files being backed up usually occupy more space than is available on one floppy disk, [Archive file] can extend across more than one volume, and split into files that are named [Archive]<Sys>.Ø1, [Archive]<Sys].Ø2, etc. (with file extensions supplied by the Backup Volume command).

If a volume name other than Archive is specified, you must also specify a directory name, so that the addition of a sequence number produces a valid file specification. That is, specify [Xyz]<Sys> to get [Xyz]<Sys>.Ø1, [Xyz]<Sys>.Ø2, etc., or [Xyz]<Sys>Abc to get [Xyz]<Sys>Abc.Ø1, [Xyz]<Sys>Abc.Ø2, etc.

[Delete existing archive file?]   Default: No

If you enter "yes", any existing archive file is automatically overwritten.

If you enter "no" or leave the field blank, and the archive file already exists, Backup Volume prompts:

```
File already exists.  Delete?
(Press GO to confirm, CANCEL to deny,
or FINISH to return to the Executive.)
```

[Log file]   Default: Screen display only

Enter the name of the file to which you wish to write a report of the backup.

If the log file exists, the log is appended to it. If it does not exist, it is created.

If no log file is named, the log appears only on the screen.

[Display structures?] Default: No

If you enter "yes", Backup Volume gives a detailed analysis of the volume control structures. This should be used only by system programmers for file system error analysis.

Description

Operation. Backup Volume is intended for use by system administrators. It displays passwords for all directories and files. In installations where it is desirable to protect the file system, the volume should be password protected.

Backup Volume is able to recover all the file information from a volume (even a scrambled one) by accessing only one of the two Volume Home Blocks and one of each pair of File Header Blocks.

Initializing Floppy Disks. Before beginning a backup, use the IVolume command to initialize a supply of floppy disks. All these floppy disks must have the same volume name and no password. (Initializing floppy disks for Backup Volume is described in the "Volume Management" subsection of Section 5, "Advanced Concepts.") The default name for the archive file is Archive. Be sure to start with enough initialized floppy disks to hold all the files being backed up. You cannot stop in the middle to initialize more disks.

Volume Backup. Backup Volume verifies the consistency of the specified parameters and opens the log file (if one is specified). Backup Volume prompts you to mount the first volume of the archive file. For example, if [Archive file] is defaulted, Backup Volume prompts you with this message:

```
Please mount [Archive]<Sys>.Ø1
(Press GO to confirm, CANCEL to deny,
or FINISH to return to the Exective.)
```

Mount the appropriate volume and press GO to confirm. If the volume you are trying to back up is in use (either as a system volume or by another user at a cluster workstation), Backup Volume displays this warning:

```
Device is currently in use.
(Press GO to confirm, CANCEL to deny,
or FINISH to return to the Exective.)
```

During a backup, it is best to restrict access to the volume. Otherwise you cannot be certain that file modifications made by other users are included in the backup. Also, when access is not restricted, the verification pass may indicate nonexistent consistency errors.

Even when Backup Volume is running, users can still access the volume, since Backup Volume does not dismount it or take it offline.

Method. Unless specified otherwise, Backup Volume first performs a backup pass, backing up each file. It reports any errors found in the backup pass. It then performs a verification pass to verify the integrity of all volume control structures. It reports any irregularities to the screen and, if specified, in the log file.

After Backup Volume fills one disk, it prompts you to mount the next:

```
Please mount [Archive]<Sys>.Ø2
(Press GO to confirm, CANCEL to deny,
or FINISH to return to the Executive.)
```

Overwriting Existing Archive Files. If an archive file of the same name already exists on the archive file (most probably from a prior backup), and "yes" was not specified in [Delete existing archive file?], Backup Volume asks if the file is to be overwritten. If CANCEL or FINISH is pressed, Backup Volume terminates and returns to the Executive. When overwriting archive files, it is important that they be overwritten in sequence. Otherwise, Backup Volume does not recognize that the archive file on the floppy disk can be overwritten.

Incremental Backup. If a date and time are specified, only those files modified on or after that date and time are backed up. This is termed an incremental backup.

Here is a recommended method for performing incremental backup.

Prepare three sets of archive floppy disks (sets A, B, and C). A and B, each consisting of 25 disks, are for full backups on alternate weeks. Set C is divided into four subsets (1-4) for the incremental backups. Determine how large set C must be, depending on the number of files changed daily.

Perform a full backup of the Winchester disk each Monday, alternating each week between set A and B. On Tuesday through Friday, perform incremental backups for all files modified since Monday, using respectively, subsets 1 through 4. The number of disks grows each day. On the next Monday, do a full backup, using the alternate set, A or B. Starting on Tuesday, reuse set C for incremental backup.

On any given day, there is always a daily record of one week's work. On Wednesday, for example, when the backup is done on subset 2, subset 3 still contains the incremental backup from the previous Thursday. There are still the two full backup sets, one containing all of last week's work, and one containing all of the previous week's.

Should the file system become scrambled or files unintentionally deleted, all the information needed to restore the entire Winchester disk is still available. Reinitialize the disk with the IVolume command, and restore the most recent full backup and the most recent incremental backup.

Limitations. There are a few limitations to incremental backups. The Restore command creates directories (if they do not already exist) for the files it is restoring; however, it does not expand or contract existing directories. Therefore, Restore may be unable to restore some files because the directories are too small. For this reason, perform a full rather than an incremental backup whenever the size of any directory is increased.

The incremental backup also cannot reflect files deleted during the week. Suppose that a file is created on Tuesday, backed up, and then deleted on Wednesday. A restoration on Thursday would include this obsolete file. Creating and then deleting many files could tie up an excessive number of archive floppy disks, and it might be better to perform a full backup more frequently than once a week.

The Winchester disks should be reinitialized with the IVolume command periodically and a full restoration performed. This prevents the file system from becoming overly fragmented, which could degrade performance.

Error Recovery. Backup Volume performs extensive checks on all volume control structures. Irregularities such as Allocation Bit Map inconsistencies, a File Header Block (FHB) without directory entries or vice versa, and invalid FHBs are reported. Irregularities within a volume have several causes. In any case, the volume must be reinitialized with the IVolume command immediately after the Backup Volume to restore the files from the archive file.

Backup Volume recovers from most errors. It needs only one good Volume Home Block (VHB) and one good FHB for each file. If a sector is unreadable, Backup Volume lists it on the screen. Status information is written to the archive file to indicate that the sector was unreadable. This information is reported when the file is restored.

Backup Volume attempts to salvage any scrambled files. If the primary FHB is scrambled or inaccessible, it uses the secondary FHB to back up the file.

Recovering an Unmountable Disk. If the operating system cannot be bootstrapped from the Winchester disk, bootstrap it from a floppy disk. Either restore the Winchester disk from the previous day's backup (if available), or use Backup Volume to remove the data from the Winchester disk and then reinitialize it with the IVolume command.

If a volume is so scrambled that it cannot be recognized by the operating system, the device name and device password must be provided for Backup Volume.

Duplicate Backup. Backing up onto duplicate sets of floppy disks provides vastly improved data reliability. The Restore command provides the capability of recovering all data from duplicate sets of backup floppy disks, even when there are nonrecoverable errors in both sets (as long as the errors do not occupy corresponding sectors on the two sets).

Example

The form below backs up and verifies a disk named "Win" onto the archive file [Archive]<Sys>.Ø1. After the backup pass is complete, the volume is verified for File Header Block, Allocation Bit Map, and directory consistency.

```
Backup Volume
  Volume or device name                          Win_____________
  [Volume or device password]                    Win_____________
  [Incremental from
   (e.g. Mon Jun 1 1981 8:ØØ pm)]                ________________
  [Suppress backup?]                             ________________
  [Suppress verification?]                       ________________
  [Archive file]                                 ________________
  [Delete existing archive file?]                ________________
  [Log file]                                     ________________
  [Display structures?]                          ________________
```

Introduction

The Batch command enters batch jobs into the system.

The Batch command processes the specified JCL file in the primary application partition if no batch queue is specified. If a batch queue is specified, Batch queues the specified JCL file for processing in the specified batch scheduling queue.

This command is implemented using the Run file Batch.Run.

Command Form

Batch
JCL file ______________________
[Parameters] ______________________
[Batch queue] ______________________
[After date/time] ______________________
[Priority] ______________________

Parameter Fields

JCL file — Enter the name of the batch JCL file you wish to process.

[Parameters] — Default: No parameters

Enter the parameters you want to pass to the specified JCL file.

[Batch queue] — Default: Primary Partition

Enter the name for the batch scheduling queue into which a queue entry is placed that describes the specified JCL file. The default is to process the JCL file in the primary application partition.

[After date/time] — Default: Current

Enter the earliest date/time the job can be processed. This allows you to submit a job to be run overnight, etc. This is applicable only in the secondary application partition.

[Priority] Default: 5

Enter the priority at which you want the generated queue entry placed in the batch scheduling queue. The possible values range from Ø (high) to 9 (low). The default is 5. This is applicable only in the secondary application partition.

Introduction

The Batch Status command can monitor the progress of job, submit or cancel a job, remove a batch partition, or print a batch manager's log file. It can operate either on a batch manager or on a batch job being processed by a batch manager executing in any workstation in the cluster configuration.

This command is implemented using the Run file Batch.Run.

Description of Batch Status Displays

The Batch Status command consists of a series of the following displays:

- Main display, which shows the status of all the batch managers in the operating system
- Job display, which shows the status of a group of batch jobs
- Queued Job display, which shows the status of a single queued job
- Executing Job display, which shows the status of a single executing job

The Main display consists of a set of entries that are batch managers; the Job, Executing Job, and Queued Job displays consist of a set of entries that are batch jobs. Each display has associated with it a set of functions. The user can monitor the status of the entries on one of the display's entries.

Modes. Batch Status can execute in either of two modes: monitor or operational. The current mode is shown at the upper left corner of the four displays. The user can switch modes at any time by using the TAB, Left Arrow, or Right Arrow key to move the selection to "Monitor Mode" or "Operational Mode" and pressing GO. Batch Status must be in operational mode for diplays to be changed.

Monitor mode is used to monitor the progress of one job, several jobs, or of a batch manager. In monitor mode, a display is updated every 3Ø seconds. Only one page of entries is shown.

Operational mode is used to execute a function, such as canceling a job, submitting a job, or printing a batch manager's log file.

To execute one of the functions, you must first select an entry (batch manager or batch job) on a display on which to operate.

The selected entry is indicated by the highlighted field. You can press the NEXT, RETURN, up arrow, or down arrow key to change the selection.

After selecting the entry, you have two ways of executing the desired function. The function can be selected by using the TAB, right arrow, or left arrow key to move the highlighted box to the desired function. Then, to execute the function, press GO. Or you can simply type the single letter located beneath the desired function.

Batch status displays are not automatically updated in operational mode. However, you can manually update them with the PREV PAGE key. If there is more than one page of entries, you can press NEXT PAGE to page through succeeding ones.

Main Display. When Batch Status in invoked, the Main display appears on the screen. The Main display shows the current status of all the batch managers in the system and the jobs they are processing. From this display, you can submit a batch job, query the status of a previously submitted job, or cancel a previously submitted job. You can also print a batch manager's log file or remove the batch partition in which a batch manager is executing. The main Batch Status display is shown in Figure 6-1.

Job Display. The Job display shows the status of a group of jobs in a single batch queue. This display is accessed from the Main display when the Query Job Status function is executed. (See the "Query Job Status" subsection below.) The Job display shows the following information:

o the status of all jobs in the specified queue

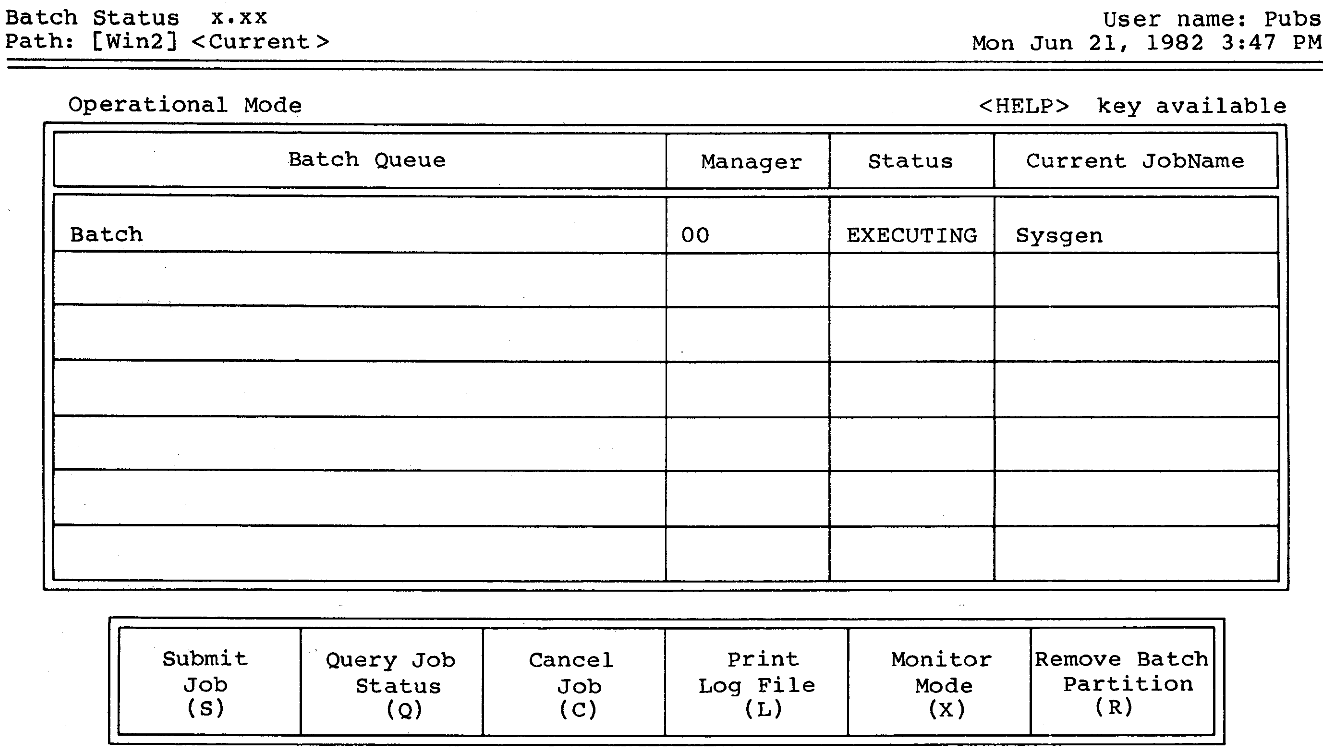
- the status of all jobs with a specified job name in the specified queue submitted by any user

- the status of all jobs submitted by a specified user in the specified queue

From this display, you can cancel a submitted job or display the status of a single job. You can return to the Main display by selecting the Main display function or by pressing CANCEL. The Job display is shown in Figure 6-2.

Queued Job Display. The Queued Job display shows the status of a single queued job. It is accessed from the Main display or from a Job display. The displayed information includes the name of the JCL file, the time the job was submitted, and the position in the queue. From this display, you can cancel the job. You can return to the previous diplay by selecting the Main display function or by pressing CANCEL. The Queued Job display is shown in Figure 6-3.

Executing Job Display. The Executing Job display shows the status of a single executing job. It is accessed from the Main display or from a Job display. The displayed information includes the time the job was started and the current job step being executed. From the display, you can cancel the job or cancel the current job step. You can return to the previous display by selecting the Main display function or by pressing CANCEL. The Executing Job display is shown below in Figure 6-4.

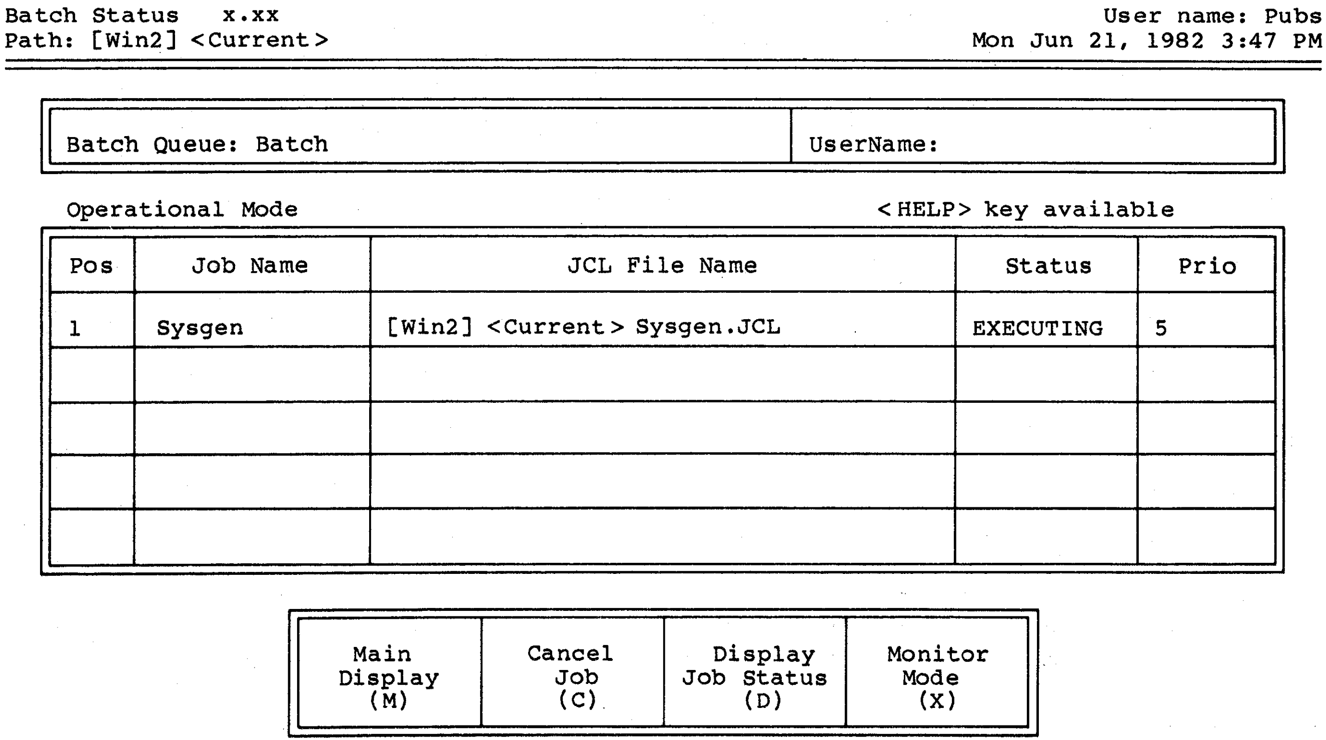


```
Batch Status  x.xx                                User name: Pubs
Path: [Win2] <Current>                 Mon Jun 21, 1982 3:47 PM
```

Operational Mode <HELP> key available

Batch Queue	Manager	Status	Current JobName
Batch	00	EXECUTING	Sysgen

Figure 6-1. Batch Status Main Display.



```
Batch Status   x.xx                                        User name: Pubs
Path: [Win2] <Current>                            Mon Jun 21, 1982 3:47 PM
```

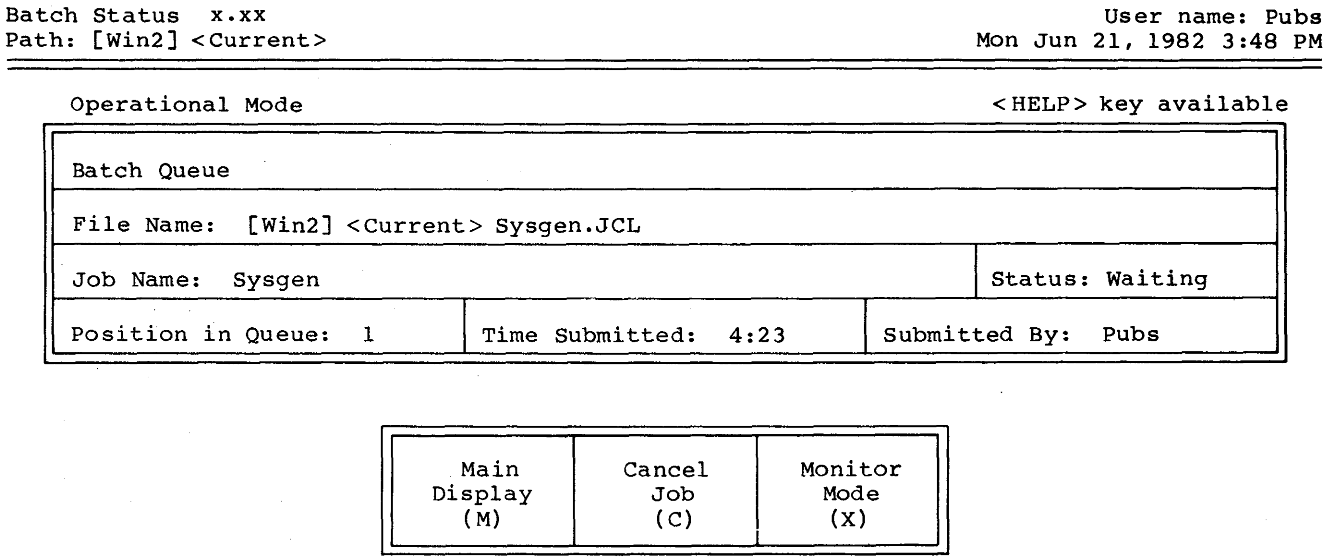
Batch Queue: Batch	UserName:

Operational Mode   <HELP> key available

Pos	Job Name	JCL File Name	Status	Prio
1	Sysgen	[Win2] <Current> Sysgen.JCL	EXECUTING	5

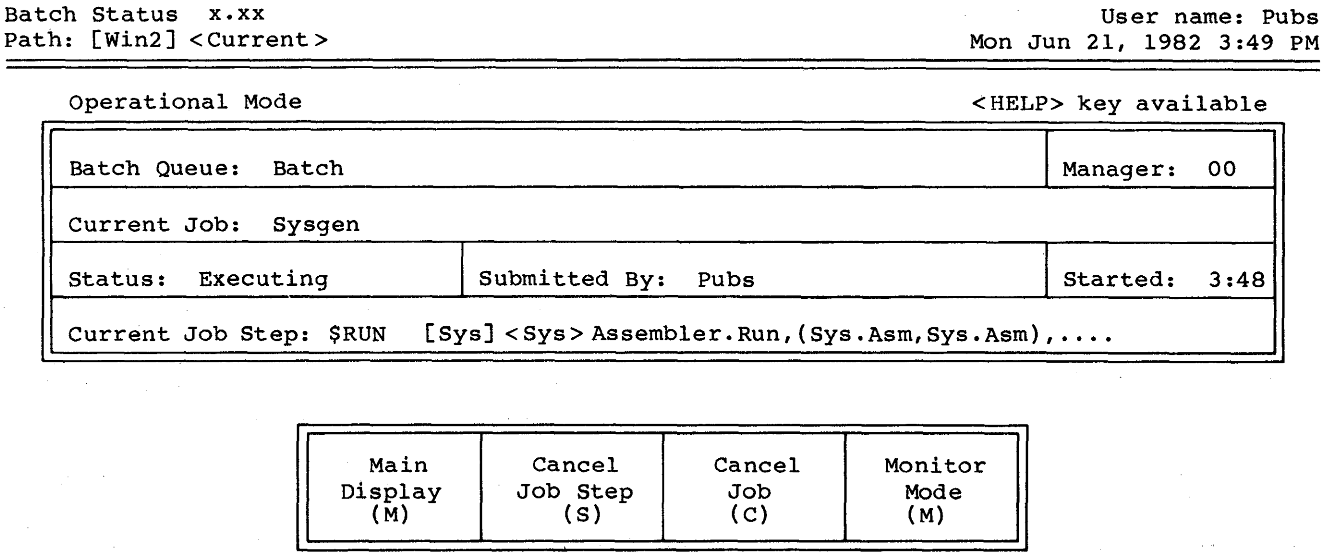
Main Display (M)	Cancel Job (C)	Display Job Status (D)	Monitor Mode (X)

Figure 6-2. Batch Status Job Display.



```
Batch Status  x.xx                                      User name: Pubs
Path: [Win2] <Current>                         Mon Jun 21, 1982 3:48 PM

   Operational Mode                               <HELP> key available

   Batch Queue

   File Name:  [Win2] <Current> Sysgen.JCL

   Job Name:  Sysgen                               Status: Waiting

   Position in Queue:  1   Time Submitted:  4:23   Submitted By:  Pubs
```

Figure 6-3. Batch Status Queued Job Display.



```
Batch Status  x.xx                                        User name: Pubs
Path: [Win2]<Current>                              Mon Jun 21, 1982 3:49 PM

  Operational Mode                                   <HELP> key available

  Batch Queue:  Batch                                      Manager:  00

  Current Job:  Sysgen

  Status:  Executing     Submitted By:  Pubs               Started:  3:48

  Current Job Step: $RUN  [Sys]<Sys>Assembler.Run,(Sys.Asm,Sys.Asm),....
```

Figure 6-4. Batch Status Executing Job Display.

Batch Status Subcommands

Batch status has six subcommands:

- Cancel Batch Job
- Cancel Job Step
- Print Log File
- Query Job Status
- Remove Batch Partition
- Submit Batch Job

Each subcommand is described below.

Cancel Batch Job

The Cancel Batch Job subcommand cancels the specified batch job. The job will be canceled regardless of whether it is executing or queued. If more than one job with the same name is ubmitted to the same batch scheduling queue by the same user, the first job in the queue is canceled.

Subcommand Form

Cancel Batch Job
  Batch queue ________________________________
  JobName ________________________________
  UserName ________________________________

Subcommand Parameter Fields

Batch queue

Enter the batch scheduling queue in which the job is queued.

JobName — Enter the job name in the $JOB statement of the submitted JCL file.

UserName — Enter the logged-in user name when the job was submitted.

Cancel Job Step

The Cancel Job Step subcommand cancels the specified job step. If the specified job step finishes before Cancel Job Step is issued, no action is be taken.

Subcommand Form

Cancel Job Step
  Batch queue ______________________________
  JobName ______________________________
  UserName ______________________________

Subcommand Parameter Fields

Batch queue

Enter the batch scheduling queue in which the job is queued.

JobName	Enter the job name in the $JOB statement of the specified job.
UserName	Enter the current job step being executed.

Print Log File

The Print Log File subcommand prints the log file of the specified batch manager. The specified batch manager copies the log file to a Printer Spooler file named QQQQNN.Log, where QQQQ is the batch scheduling and NN is the batch manager identification number. The Printer Spooler file is queued for printing in the SPL printer spooler scheduling queue.

Subcommand Form

Print Log File
Batch Manager ______________________________
Batch queue ______________________________
[Truncate log file?] ______________________________

Subcommand Parameter Fields

Batch Manager

Enter the identification number of the specified batch manager.

Batch queue

Enter the name for the batch scheduling queue being served by the batch manager.

[Truncate log file?] Default: No

If you enter "yes", the batch manager truncates its log file after printing a copy.

Query Job Status

The Query Job Status subcommand displays the status of the specified job(s).

The status of a single job can be displayed by filling in all fields.

The status of all jobs in a specified queue submitted by a specified user can be displayed by filling in the Batch queue and UserName fields and specifying an asterisk in the JobName field.

The status of all jobs of a specified name in a specified queue submitted by any user can be displayed by filling in the Batch queue and JobName fields and specifying an asterisk in the UserName field.

The status of all jobs in a specified queue can be displayed by filling in the Batch queue field and specifying an asterisk in both the JobName and the UserName fields.

Subcommand Form

Query Job Status
Batch queue ______________________
JobName (* for all jobs) ______________________
UserName (* for all users) ______________________

Subcommand Parameter Fields

Batch queue (* for all jobs)

Enter the name for the batch scheduling queue in which the job is queued.

JobName (* for all jobs)

Enter the job name in the $JOB statement of the submitted JCL file.

UserName

Enter the user name logged-in when the job was submitted.

Remove Batch Partition

The Remove Batch Partition subcommand terminates the job currently being processed and vacates the batch partition.

Subcommand Form

Remove Batch Partition
Batch Manager ______________________________
Batch queue ______________________________

Subcommand Parameter Fields

Batch Manager

Enter the identification number for the specified batch manager.

Batch queue

Enter the name of the batch scheduling queue served by the batch manager.

Submit Batch Job

The Submit Batch Job subcommand submits a batch job to a batch queue for processing in a secondary application partition. Batch Status opens the specified batch job's JCL file and makes sure it contains a valid $JOB statement. Batch Status also makes sure the user profile specified in the $JOB statement is present in the [!Sys]<Sys> directory.

Subcommand Form

Submit Batch Job
Batch queue ________________
Batch JCL file ________________
[Parameters] ________________
[After date/time] ________________
[Priority] ________________

Subcommand Parameter Fields

Batch queue

Enter the name of the batch scheduling queue in which an entry is placed that describes the submitted JCL file.

Batch JCL file

Enter the JCL file to be processed by a batch manager.

[Parameters] Default: No parameters

Enter the parameters to be passed to the specified JCL file.

[After date/time] Default: Current

Enter the earliest date/time the job can be processed. This allows you to submit a job to be run overnight, etc.

[Priority] Default: 5

Enter the priority at which you want the queue entry placed in the batch scheduling queue. The possible values range from Ø to 9, with Ø the highest priority. The default is 5.

Introduction

The Bootstrap command uses the operating system to load a program into memory, replaces the operating system with that program, and then transfers control of the workstation to that program. The program can be a diagnostic or another operating system.

Bootstrap can also be invoked by the operating system during initialization. In this case, there are no parameters from the Executive. (See the subsection "Bootstrapping", in Section 5, "Advanced Concepts.")

This command is implemented using the Run file Bootstrap.Run.

Command Form

Bootstrap
File name ________________________________

Parameter Fields

File name	Enter the name of the file you wish to bootstrap (overwriting in memory the operating system currently running at the workstation).

Introduction

The Change Volume Name command changes a volume's name and/or password. You can also add or delete volume passwords with this command.

Change Volume Name can only be invoked for disks that are not currently in use (that is, a disk with no files currently open). To change the volume name of the system disk, bootstrap the operating system from another disk. (See the Bootstrap command.)

This command is implemented using the Run file ChgVolName.Run.

Command Form

Change Volume Name
Device name ____________________
[Device password] ____________________
[Old volume password] ____________________
New volume name ____________________
[New volume password] ____________________

Parameter Fields

Device name

Enter the name of the disk device on which the volume is mounted. This name is assigned at system build. Examples of device names are FØ, F1, DØ, and D1.

[Device password] Default: No password used

Enter the the password assigned the device at system build.

For the version of the CTOS operating system on the distribution floppy disk, the floppy disk drives have no passwords and the Winchester disk drives have the passwords: DØ, D1, and D2.

If no password was assigned, leave this blank.

[Old volume password]   Default: No password used

Enter the password assigned the volume with the IVolume command.

If you wish to change only the volume name, the old volume password must still be specified.

If the volume was not assigned a password when it was IVolumed, leave this field blank.

New volume name

Enter the new name you want to assign to this volume. Volume names may have up to 12 characters.

This field is NOT optional.

[New volume password]   Default: No new password

Enter the new password you want to assign to this volume. Passwords may have up to 12 characters.

If no new volume password is specified, a new one is not assigned.

Example

To rename the volume mounted on FØ to Source with no password, type

```
Change Volume Name
  Device name                FØ________________________
  [Device password]          __________________________
  [Old volume password]      __________________________
  New volume name            Source____________________
  [New volume password]      __________________________
```

CLUSTER STATUS

Introduction

The Cluster Status command displays status information about the activity on a cluster system for all lines and for a specified communications line.

This command is implemented using the Run file ClusterStatus.Run.

Command Form

Cluster Status
[Line number] ______________________________
[One time display?] ______________________________

Parameter Fields

[Line number] Default: Overview of all lines

Enter the number of the line, Ø through 4, for which status information is requested. Only one line can be specified. If you enter a number here, you get the detailed display.

If you leave this field blank, the overview display shows you which users are on each line of your cluster.

[One time display?] Default: No

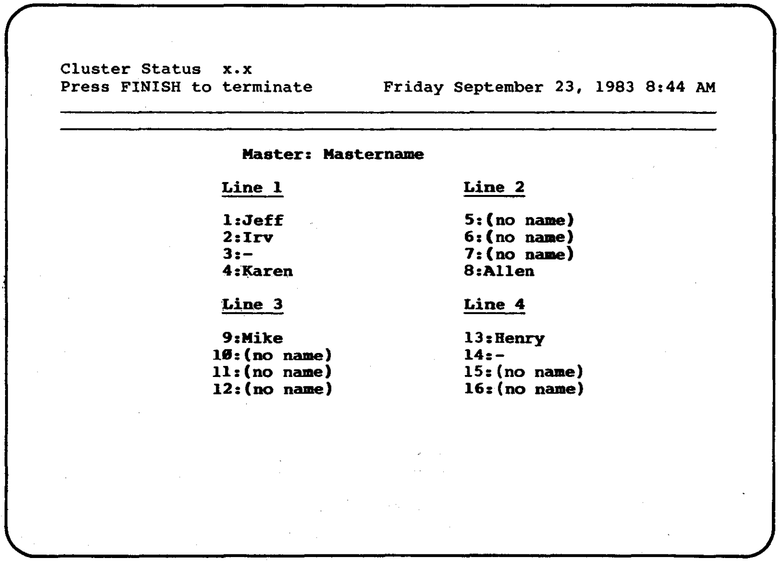
If you enter "yes", the Executive command prompt is automatically returned to the screen after the requested status display appears.

If you enter "no" or leave this field blank, the status display remains on the screen, and is updated every second. You can return to the Executive at any time by pressing FINISH.

Description

An overview display (Figure 6-5) shows the Master name, the cluster lines configured, which workstations are active, and the user name given at SignOn for each active workstation.

If the current user on the workstation signed on using the default User Configuration file (.User), the display indicates this by displaying "(no name)". If no workstation is attached to the line, this is indicated with a dash. This display is the default and can be accessed by typing "Cluster Status" and pressing GO.



```
Cluster Status  x.x
Press FINISH to terminate        Friday September 23, 1983 8:44 AM
________________________________________________________________
________________________________________________________________

                  Master: Mastername

                 Line 1                   Line 2

                 1:Jeff                   5:(no name)
                 2:Irv                    6:(no name)
                 3:-                      7:(no name)
                 4:Karen                  8:Allen

                 Line 3                   Line 4

                 9:Mike                   13:Henry
                1Ø:(no name)              14:-
                11:(no name)              15:(no name)
                12:(no name)              16:(no name)
```

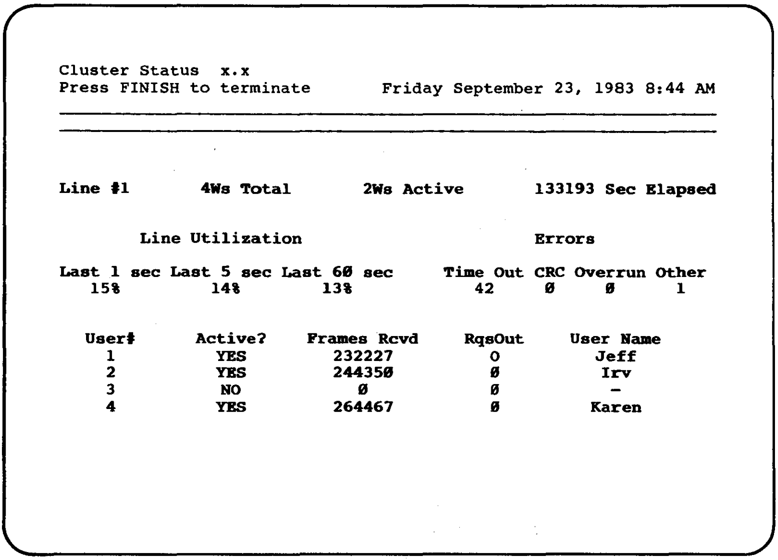
Figure 6-5. Cluster Status Overview Display.

A more detailed display (Figure 6-6) can be requested for each communications line that shows

- o line number
- o total number of workstations configured
- o total number of workstations active
- o time elapsed since the cluster became active
- o line utilization (percent of time busy)
- o the number of errors encountered for each type of error listed
- o for each active workstation,
  - workstation number
  - whether the workstation is active
  - number of request frames received since the user became active
  - number of outstanding requests
  - user name

The communications lines are identified as follows:

Line Number	Communications Line
Ø	Standard channel
1	CommIOP 1, channel A
2	CommIOP 1, channel B
3	CommIOP 2, channel A
4	CommIOP 2, channel B



```
Cluster Status  x.x
Press FINISH to terminate       Friday September 23, 1983 8:44 AM
_________________________________________________________________
_________________________________________________________________


Line #1       4Ws Total       2Ws Active       133193 Sec Elapsed

        Line Utilization                       Errors

Last 1 sec Last 5 sec Last 6Ø sec    Time Out CRC Overrun Other
   15%        14%         13%           42     Ø     Ø      1

  User#       Active?     Frames Rcvd     RqsOut     User Name
    1           YES         232227          O          Jeff
    2           YES         24435Ø          Ø          Irv
    3           NO            Ø             Ø           -
    4           YES         264467          Ø          Karen
```

Figure 6-6. Cluster Status Detailed Display.

COPY

Introduction

The Copy command copies the contents of a file to another file. If the file you wish to copy to already exists, its contents will be replaced. To add the contents of a file to an existing file, use the Append command.

The Copy command can also be used to copy the contents of a file to a printer by direct printing. See the subsection "File Management" in Section 4, "Basic Concepts" for details.

If wild cards are used with the Copy command, every file matching the specification given in the File from field is copied to a file named according to the specification given in the File to field. The File from field must contain the same number of wild card characters as the File to field and they must be in the same order.

This command is implemented internally by the Executive using the value !2.

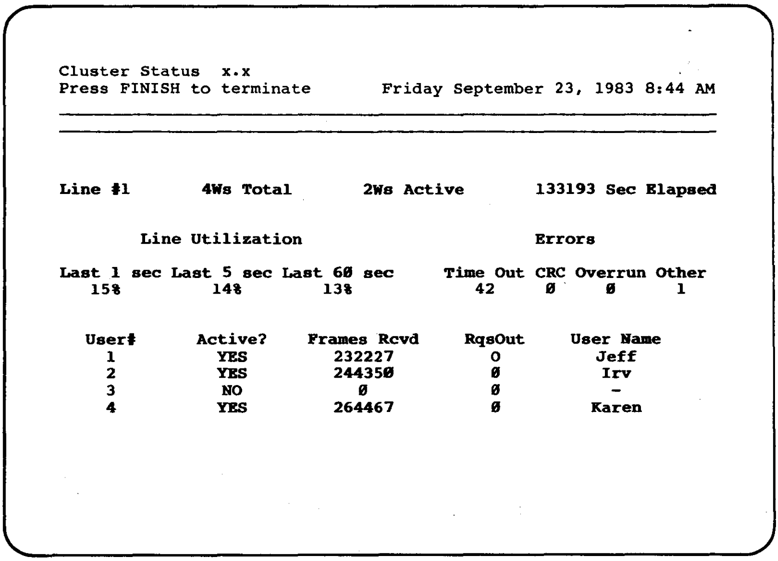
Command Form

```
Copy
  File from             ______________________________
  File to               ______________________________
  [Overwrite ok?]       ______________________________
  [Confirm each?]       ______________________________
```

Parameter Fields

File from — Enter a file specification for the file(s) you wish to copy. If you use wild card characters in the file specification you must also use them when you specify a file in the File to field.

File to — Enter a file specification for the new file(s) to which you want to copy the File from field entries.



```
Cluster Status  x.x
Press FINISH to terminate       Friday September 23, 1983 8:44 AM
_________________________________________________________________
_________________________________________________________________


Line #1       4Ws Total       2Ws Active       133193 Sec Elapsed

        Line Utilization                           Errors

Last 1 sec Last 5 sec Last 6Ø sec     Time Out CRC Overrun Other
   15%         14%         13%            42     Ø     Ø       1

  User#       Active?     Frames Rcvd     RqsOut     User Name
    1           YES         232227          O          Jeff
    2           YES         24435Ø          Ø          Irv
    3           NO            Ø             Ø           -
    4           YES         264467          Ø          Karen
```

Figure 6-6. Cluster Status Detailed Display.

COPY

Introduction

The Copy command copies the contents of a file to another file. If the file you wish to copy to already exists, its contents will be replaced. To add the contents of a file to an existing file, use the Append command.

The Copy command can also be used to copy the contents of a file to a printer by direct printing. See the subsection "File Management" in Section 4, "Basic Concepts" for details.

If wild cards are used with the Copy command, every file matching the specification given in the File from field is copied to a file named according to the specification given in the File to field. The File from field must contain the same number of wild card characters as the File to field and they must be in the same order.

This command is implemented internally by the Executive using the value !2.

Command Form

Copy
  File from ______________________________
  File to ______________________________
  [Overwrite ok?] ______________________________
  [Confirm each?] ______________________________

Parameter Fields

File from — Enter a file specification for the file(s) you wish to copy. If you use wild card characters in the file specification you must also use them when you specify a file in the File to field.

File to — Enter a file specification for the new file(s) to which you want to copy the File from field entries.

[Overwrite ok?]   Default: Ask for Confirmation

Enter "yes", "no", or leave the field blank.

If you enter "yes" and the file named in the File to field already exists, then the contents of the existing file are deleted and the copy operation continues.

If you enter "no" and the file named in the File to field already exists, the file will not be copied.

If you leave the field blank, the Executive will prompt you for confirmation before deleting the contents of the existing file. You can confirm this by pressing GO, deny it by pressing CANCEL, or stop copying files by pressing FINISH.

[Confirm each?]   Default: No

If you enter "yes", then you are prompted to confirm each copy operation by pressing GO, deny it by pressing CANCEL, or stop copying files by pressing FINISH.

If you enter "no" or leave the field blank, then you are not prompted for confirmation and the copy operation is performed.

Examples

The following form shows how you use the Copy command.

```
Copy
  File from                abc____________________
  File to                  def____________________
  [Overwrite ok?]          _______________________
  [Confirm each?]          _______________________
```

The following is a example of the use of a wild card character with the Copy command. If the currently logged-in volume and directory contains the files

```
    memo.Ø2-16-8Ø
    memo.Ø2-3Ø-8Ø
    memo.Ø5-Ø2-8Ø
```

then execution of the form

```
    Copy
      File from              memo.Ø2-*-8Ø__________
      File to                Feb-*-8Ø______________
      [Overwrite ok?]        ______________________
      [Confirm each?]        ______________________
```

copies the files memo.Ø2-16-8Ø and memo.Ø2-3Ø-8Ø to the files Feb-16-8Ø and Feb-3Ø-8Ø.

CREATE CONFIGURATION FILE — Advanced Command

Introduction

The Create Configuration File command creates a file called a configuration file, which specifies device-specific parameters for a device interface.

Four device types can be configured:

- communications lines ([Comm]A and [Comm]B)
- spooled printers ([Spl])
- serial printers ([Ptr]A and [Ptr]B)
- parallel printers ([Lpt])

The Create Configuration File command form prompts you for the type of device. After you fill in the form and press GO, a Subcommand form will be displayed appropriate for the type of configuration file you wish to create. The Subcommand form is used to specify the device characteristics.

This command is implemented using the Run file CreateConfig.Run.

Command Form

Create Configuration File
Configuration file name ________________
Device type (comm, parallel lpt, or serial ptr) ________________

Parameter Fields

Configuration file name

Enter the name you want to assign to the configuration file or the name of the configuration file you want to modify.

Device type (comm, parallel lpt, or serial ptr)

Enter the code for the type of device the configuration file will be used for. Use c, p, or s for communications, a parallel printer, or a serial printer.

After this form is completed and you press GO, the appropriate subcommand (Communications Characteristics, Parallel Printer Characteristics, or Serial Printer Characteristics) appears.

Description

Each device has an associated default configuration file provided with the standard software. Table 6-1 lists the file specifications for default configuration files for each device. Table 6-2 below summarizes the characteristics of the default configuration files.

Table 6-1. Default Configuration File Specifications.

Device Specification	Default Configuration File Specification
[Comm]A	[Sys]<Sys>CommAConfig.Sys
[Comm]B	[Sys]<Sys>CommBConfig.Sys
[Spl]	[Sys]<Sys>SplConfig.Sys
[Ptr]A	[Sys]<Sys>PtrAConfig.Sys
[Ptr]B	[Sys]<Sys>PtrBConfig.Sys
[Lpt]	[Sys]<Sys>LptConfig.Sys

Characteristics in an existing configuration file can be changed if you specify its name and device type in the Create Configuration File form.

A particular configuration file can be specified when the OpenByteStream operation is called for communications or printer byte streams. This is done by appending "&" and the full file specification for the configuration file to the device specification.

Example: [LPT]&[Volname]<Dirname>filename.

If a configuration file is not specified in this way, the default file is used.

For information about communications and printer byte streams, and the OpenByteStream operation, see the section on the Sequential Access Method in the CTOS Operating System Manual.

Table 6-2. Characteristics of Default Configuration Files.

	Device Specification		
Characteristics	[Comm]A/B	[Spl] [Lpt]	[Ptr]A/B
Data bits	7	na	7
Parity	Ø	na	even
Baud rate	96ØØ	na	12ØØ
Stop bits	1	na	1
Transmit time out	no time out		
Receive	no time out	na	na
CR/LF mapping mode	new line	na	na
New line mapping mode	RETURN (ØAh) to ASCII CR/LF		
Line control mode	XON/XOFF	na	XON/XOFF
EOF byte	Ø4 (EOT)	na	na
Expand tab size	na	8	8
Number of characters per line	na	132	132
ACK delay	na	Ø	na

na = not applicable

Subcommand Form

Communications Characteristics

[Data bits (5, 6, 7, or 8; default = 7)] ______________

[Parity (none, even, odd, Ø, or 1; default = Ø)] ______________

Baud rate (up to 192ØØ; default = 96ØØ)] ______________

Stop bits (1 or 2; default = 1)] ______________

Transmit time out (number of seconds; default = no time out)] ______________

[Receive time out (number of seconds; default = no time out)] ______________

[CR/LF mapping mode (binary or new line; default = new line)] ______________

[New line mapping mode (binary, CR, or CR/LF; default = CR/LF)] ______________

[Line control mode (none, XON/XOFF, CTS, or both; default = XON/XOFF)] ______________

EOF byte (hexadecimal value or none; default = Ø4)] ______________

Subcommand Parameter Fields

[Data bits (5, 6, 7, or 8; default = 7)]

Enter the number of data bits per character.

Data bits does not include the parity bit if parity is even, odd, Ø, or 1.

[Parity (none, even, odd, Ø, or 1; default = Ø)]

Enter the state of the parity bit as a check on the data bits. When specifying "none", increase the number of data bits (see above) to account for the absence of a parity bit.

[Baud rate (up to 192ØØ; default = 96ØØ)]

Enter the baud rate. Baud rate is a transmission speed from 2Ø to 192ØØ. Common baud rates are 11Ø, 15Ø, 3ØØ, 12ØØ, 24ØØ, 48ØØ, and 96ØØ.

[Stop bits (1 or 2; default = 1)]

Enter the number of stop bits per character.

[Transmit time out (number of seconds; default = no time out)]

Enter the number of seconds a write operation waits to begin transmitting a character before returning status code 3ØØ ("Device not ready").

[Receive time out (number of seconds; default = no time out)]

Enter the number of seconds a read operation waits without receiving a character before returning status code 6Ø2 ("No character available").

If Ø is specified and no character is currently available, status code 6Ø2 ("No character available") is immediately returned from a read operation.

If this field is defaulted, status code 6Ø2 ("No character available") is never returned from a read operation.

[CR/LF mapping mode (binary or new line; default = new line)]

Enter "Binary" or "New Line" This field determines the mapping of incoming carriage returns (CR) and linefeeds (LF).

Binary does not map any incoming CRs or LFs.

New line maps an incoming single CR, single LF, or CR/LF combination into a carriage RETURN (ØAh).

[New line mapping mode (binary, CR, or CR/LF; default = CR/LF)]

Enter "binary", "CR", or "CR/LF". This field determines the mapping of outgoing carriage RETURNs (ØAh).

Binary does not map any outgoing carriage RETURNs.

CR maps an outgoing carriage RETURN into an ASCII CR (ØDh).

CR/LF maps an outgoing carriage RETURN into a carraige return linefeed sequence (ASCII CR/LF (ØDh/ØAh)).

[Line control mode (none, XON/XOFF, CTS, or both; default = XON/XOFF)]

Enter "none", "XON/XOFF", "CTS", or "both". This field determines the line control mode to be used by the receiving device to control the flow of data.

"None" selects no line control.

"XON/XOFF" selects suspension of transmission from the time of receipt of an XOFF control character (13h) until the receipt of an XON control character (11h).

"CTS" selects suspension of transmission when the Clear to Send signal is not received.

"both" suspends transmission if the XOFF control character is received or if the Clear to Send signal is not received.

[EOF byte (hexadecimal value or none; default = Ø4)]

Enter the hexadecimal value of the byte to compare with the received byte to detect the end of the file.

If "none" is specified, status code 1 ("End of file") is never returned from a communications byte stream.

Ø4 is the ASCII code for the EOT (End of Transmission) character.

Subcommand Form

Parallel Printer Characteristics
[New line mapping mode
(binary, CR, or
CR/LF;default = CR/LF)] ________________
[Expand tab size (default = 8)] ________________
[Number of characters
per line (default = 132)] ________________
[Transmit time out
(number of seconds;
default = no time out)] ________________
[Additional ACK delay
(units of 1ØØ
microseconds; default = Ø)] ________________

Subcommand Parameter Fields

[New line mapping mode (binary, CR, or CR/LF;
default = CR/LF)]

Enter "binary", "CR", or "CR/LF". This field determines the mapping of outgoing RETURNs (ØAh).

Binary does not map any outgoing RETURNs.

CR maps an outgoing carriage RETURN character into a ASCII CR (ØDh).

CR/LF maps an outgoing carriage RETURN character into a carriage return line-feed sequence (ASCII CR/LF (ØDh/ØAh)).

[Expand tab size (default = 8)]

Enter the number of blanks into which to map a TAB character.

[Number of characters per line (default = 132)]

Enter the maximum number of characters in a print line.

[Transmit time out (number of seconds;
default = no time out)]

Enter the number of seconds a write operation waits to begin transmitting a character before returning status code 3ØØ ("Device not ready").

[Additional ACK delay (units of 1ØØ microseconds; default = Ø)]

For printers that use a longer acknowledgment signal (ACK) than 1Ø-2Ø microseconds, enter the additional delay.

Subcommand Form

```
Serial Printer Characteristics
  [Data bits
    (5, 6, 7, or 8; default = 7)]                 ____________
  [Parity (none, even, odd,
     Ø, or 1; default = even)]                    ____________
  [Baud rate
    (up to 192ØØ; default = 12ØØ)]                ____________
  [Stop bits (1 or 2; default = 1)]               ____________
  [Transmit time out
    (number of seconds;
    default = no time out)]                       ____________
  [New line mapping mode
    (binary, CR,
    or CR/LF; default = CR/LF)]                   ____________
  [Line control mode
    (none, XON/XOFF, CTS,
    or both; default = XON/XOFF)]                 ____________
  [Expand tab size (default = 8)]                 ____________
  [Number of characters
    per line (default = 132)]                     ____________
```

Subcommand Parameter Fields

[Data bits (5, 6, 7, or 8; default = 7)]

Enter the number of data bits per character.

Data bits does not include the parity bit if parity is even, odd, Ø, or 1.

[Parity (none, even, odd, Ø, or 1; default = even)]

Enter the state of the parity bit as a check on the data bits. When specifying "none", increase the number of data bits to account for the absence of a parity bit.

[Baud rate (up to 192ØØ; default = 12ØØ)]

Enter a transmission speed from 2Ø to 192ØØ. Common baud rates are 11Ø, 15Ø, 3ØØ, 12ØØ, 24ØØ, 48ØØ, and 96ØØ.

[Stop bits (1 or 2; default = 1)]

Enter the number of stop bits per character.

[Transmit time out (number of seconds; default = no time out)]

Enter the number of seconds a write operation waits to begin transmitting a character before returning status code 3ØØ ("Device not ready").

[New line mapping mode (binary, CR, or CR/LF; default = CR/LF)]

Enter "binary", "CR", or "CR/LF". This field determines the mapping of outgoing carriage RETURNs (ØAh).

Binary does not map any outgoing carriage RETURNs.

CR maps an outgoing carriage RETURN into an ASCII CR (ØDh).

CR/LF maps an outgoing carriage RETURN into a carriage return linefeed sequence (ASCII CR/LF (ØDh/ØAh)).

[Line control mode (none, XON/XOFF, CTS, or both; default = XON/XOFF)]

Enter "none", "XON/XOFF", "CTS", or "both". This field determines the line control mode to be used by the receiving device to control the flow of data.

"None" selects no line control.

"XON/XOFF" selects suspension of transmission from the time of receipt of an XOFF control character (13h) until the receipt of an XON control character (11h).

"CTS" selects suspension of transmission when the Clear to Send signal is not received.

"both" suspends transmission if the XOFF control character is received or if the Clear to Send signal is not received.

[Expand tab size (default = 8)]

Enter the number of blanks into which to map a TAB character.

[Number of characters per line default = 132)]

Enter the maximum number of characters in a print line.

CREATE DIRECTORY

Introduction

The Create Directory command makes a new directory on a volume (disk).

See Section 4, "Basic Concepts," for information on file organization, passwords, and protection levels.

This command is implemented internally by the Executive using the value !14.

Command Form

Create Directory
New directory name ________
[Default protection level (default 15)] ________
[Maximum number of files (default 45)] ________
[Password for new directory] ________
[Volume password] ________

Parameter Fields

New directory name

Enter the name of the directory you wish to create. If the directory is not to be created on the default volume, then enter a directory specification of the form [volumename]<directoryname>.

The default volume is shown in the Executive Status frame.

[Default protection level (default 15)]

Enter the decimal code for the protection level you wish to have the Executive assign by default to all files created within this directory. Only the protection levels unprotected (15), modify protected (5), and access protected (Ø) are valid.

[Maximum number of files (default 45)]

Enter the maximum number of files you want the directory to be able to handle. The default is 45.

When the maximum number of files is reached, you can not add more files to that directory.

The maximum number of files you assign in this field is divided by 15 to obtain the number of pages allocated to that directory.

[Password for new directory] Default: No password

Enter the password for the new directory. Passwords can have up to 12 characters.

[Volume password] Default: No volume password

Enter the password assigned to the volume on which the directory will be created. If there is no password for the volume, leave this field blank.

CREATE FILE

Introduction

The Create File command creates a new file, without defining its contents.

See Section 4, "Basic Concepts," for more information on file organization, passwords, and protection levels.

This command is implemented internally by the Executive using the value !5.

Command Form

Create File
  File name ________
  [Volume or directory password] ________
  [File password] ________
  [File protection level (default = 15)] ________
  [Size in sectors (default = Ø)] ________
  [Overwrite ok?] ________

Parameter Fields

File name

Enter the file specification you want to assign to the new file.

[Volume or directory password]Default: No password

Enter the password for the volume or directory within which you wish to create the file.

If the volume or directory is password protected you must enter a password here to create a new file. Otherwise, no password is required.

[File password]   Default: No file password

Enter the file password you wish to assign to the new file. If this field is defaulted, no password will be assigned to the file.

[File protection level (default = 15)]

Enter the decimal code for the protection level you wish to assign to the new file. All protection levels are valid.

[Size in sectors (default = Ø)]

Enter the number of sectors you wish to assign to the file at creation. If you leave this field blank, the file length will be equal to zero.

[Overwrite ok?] Default: Ask for Confirmation

Enter "yes", "no" or leave the field blank.

If you enter "yes" and the file named in the File name field already exists, the contents of the existing file are deleted.

If you enter "no" and the file named in the File name field already exists, the Create File operation is cancelled.

If you leave the field blank, the Executive prompts you for confirmation before the contents of the existing file are deleted. You can confirm it by pressing GO, deny it by pressing CANCEL, or stop creating files by pressing FINISH.

Introduction

The Create Partition command creates a vacant secondary partition.

This command is implemented using the Run file CreateP.Run.

Command Form

```
Create Partition
  [Memory size (default  = 200K)]                 ____
  [Partition name (default = Background00)]       ____
```

Parameter Fields

[Memory size (default = 200K)]

Enter the size of the partition to be created. The partition needs an additional 1.5K to store system data structures associated with it.

[Partition name (default = Background00)]

Enter the name of the partition.

If the default name is used, a partition is named Backgroundnn, where nn begins with 00. For example, if more than one secondary application partition is created with a default partition name, the first secondary application partition is named Background00, the second is named Background01, etc.

Example

To create a secondary application partition named Bruce with a size of 5120 bytes, complete the Create Partition form as follows:

```
Create Partition
  [Memory size (default = 200K)]               5K
  [Partition name (default = Background00)]    Bruce
```

DELETE

Introduction

The Delete command permanently destroys each file named in the File list field.

This command is implemented internally by the Executive using the value !3.

Command Form

Delete
File list ________________
[Confirm each?] ________________

Parameter Fields

File list — Enter a valid file specification, a file list, or an at-file name.

Be especially careful when using the wild card characters with the Delete command. You can easily delete more than you intended to. You can try your wild card character specification with the Files command first, to check that only the files you intend to specify are indeed included. It is also a good idea to use the [Confirm Each?] option with the Delete command when using a wild carded file specification.

[Confirm each?] — Default: No

If you enter "yes", you are prompted to confirm each deletion by pressing GO, deny it by pressing CANCEL, or stop deleting files by pressing FINISH.

If you enter "no" or leave the field blank, then you are not prompted for confirmation.

Introduction

The Disable Cluster command disables cluster operations. It causes the operating system in the master workstation to stop communicating with the cluster workstations and closes all files that are open for the cluster workstations. Any outstanding requests or new requests issued from a cluster workstation when cluster operations are disabled is returned with status code 6 ("Master workstation not running".)

Disable Cluster can only be run from the master workstation. This command is useful when installing new software in the cluster environment.

See the Resume Cluster command for reenabling cluster operations.

This command is implemented using the Run file DisableCluster.Run.

Command Form

Disable Cluster
[Time interval (seconds)] ______________

Parameter Field

[Time interval (seconds)] Default: Ø

Enter the length of time in seconds you want to delay before the cluster is disabled.

Description

You can also specify when the cluster operations will be disabled. After Disable Cluster has executed, any GetDateTime operation receives status code 46 ("Master workstation going down"). The seconds field in the returned date-time structure contains the time remaining until cluster operations are disabled.

The application system must take appropriate action when it receives status code 46 in response to its GetDateTime operation. For example, the Executive displays the message "Cluster going down xx", where xx is the number of seconds remaining before the cluster is disabled.

Introduction

The Dump command displays the contents of a file. Dump can display

- the contents in hexadecimal or ASCII
- the differences between two files

This command is implemented using the Run file Dump.Run.

Command Form

Dump
Input file ______________________
[Compare file] ______________________
[Output file] ______________________
[File address first] ______________________
[File address limit] ______________________

Parameter Fields

Input file

Enter the name of the file you want to display.

[Compare file] Default: No comparison is made

Enter the name of the file to be compared with the input file. The comparison (or display) is limited to the range of file addresses specified in the [File address first] and [File address limit] fields (described below).

If a compare file is named, a comparison of the differing parts of the input file and the compare file appears on the screen. It also is written to an output file if one is specified in the Output file field.

The comparison ends with the message

```
Differences encountered.
```

or

```
No differences encountered.
```

If a compare file is not named, only the content of the input file appears on the screen.

[Output file] Default: Display not saved

Enter the name of the file to which to write the comparison.

If no output file is named, the comparison appears only on the screen.

If the file specified in this field already exists, its prior contents are discarded. If it does not exist, it is created.

[File address first] Default = 0

Enter the hexadecimal number indicating the file address at which the comparison or display should begin.

If the field is left blank, or an invalid value is specified, the comparison or display begins at file address Ø.

The final "h" of the hexadecimal number, for example, 1F2Eh, is optional.

[File address limit] Default: FFFFFFFFh

Enter the hexadecimal number indicating the file address at which the comparison or display should end. The comparison or display is up to, but not including, this address.

If the number specified is greater than the address of the end of the file, the comparison or display ends at the end of the file.

If you leave this field blank, or an invalid value specified, the comparison or display ends at file address FFFFFFFFh.

The final "h" of the hexadecimal number, for example, 1F2Eh, is optional.

Examples

Figure 6-7 shows a comparison of two similar files.

Figure 6-8 shows the display of a single file.

```
ØØØ1   71 72 73 74 75 76 77 78 79 7A ØA 61 62 63 64 71   qrstuvwxyz.abcdq
ØØØ1   71 72 73 74 75 76 77 78 79 7A ØA 61 62 63 64 65   qrstuvwxyz.abcde

ØØØ2   65 66 67 68 6A 6B 6C 6D 6E 6F 7Ø 71 72 73 74 75   efghjklmnopqrstu
ØØØ2   66 67 68 69 6A 6B 6C 6D 6E 6F 7Ø 71 72 73 74 75   fghijklmnopqrstu

ØØØ5   ØA 61 62 63 64 65 66 67 68 69 77 6A 6B 6C 6D 6E   .abcdefghiwjklmn
ØØØ5   ØA 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F   .abcdefghijklmno

ØØØ6   6F 71 72 73 74 75 76 77 78 79 7A ØA 61 62 63 64   oqrstuvwxyz.abcd
ØØØ6   7Ø 71 72 73 74 75 76 77 78 79 7A ØA 61 62 63 64   pqrstuvwxyz.abcd

ØØØ9   6A 6B 6C 6D 6E 6F 7Ø 65 72 71 72 73 74 77 78 79   jklmnoperqrstwxy
ØØØ9   6A 6B 6C 6D 6E 6F 7Ø 71 72 73 74 75 76 77 78 79   jklmnopqrstuvwxy
```

Figure 6-7. Comparison of Two Similar Files.

```
ØØØØ   61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 7Ø   abcdefghijklmnop
ØØØ1   71 72 73 74 75 76 77 78 79 7A ØA 61 62 63 64 71   qrstuvwxyz.abcdq
ØØØ2   65 66 67 68 6A 6B 6C 6D 6E 6F 7Ø 71 72 73 74 75   efghjklmnopqrstu
ØØØ3   76 77 78 79 7A ØA 61 62 63 64 65 66 67 68 69 6A   vwxyz.abcdefghij
ØØØ4   6B 6C 6D 6E 6F 7Ø 71 72 73 74 75 76 77 78 79 7A   klmnopqrstuvwxyz
ØØØ5   ØA 61 62 63 64 65 66 67 68 69 77 6A 6B 6C 6D 6E   .abcdefghiwjklmn
ØØØ6   6F 71 72 73 74 75 76 77 78 79 7A ØA 61 62 63 64   oqrstuvwxyz.abcd
ØØØ7   65 66 67 68 69 6A 6B 6C 6D 6E 6F 7Ø 71 72 73 74   efghijklmnopqrst
ØØØ8   75 76 77 78 79 7A ØA 61 62 63 64 65 66 67 68 69   uvwxyz.abcdefghi
ØØØ9   6A 6B 6C 6D 6E 6F 7Ø 65 72 71 72 73 74 77 78 79   jklmnoperqrstwxy
ØØØA   7A ØA                                             z.
```

Figure 6-8. Display of a Single File.

FILES

Introduction

The Files command displays on the screen a list of files, optionally displays information about the files, and optionally writes the display to another file or to the printer.

This command is implemented using the Run file Files.Run.

Command Form

Files
  [File list] ______________________
  [Details?] ______________________
  [Print file] ______________________

Parameter Fields

[File list] Default: *

Enter a file specification defining the file or list of files you wish to see displayed.

You can use the wild card character in this field to specify all files, or a subset of the total files listing.

The default setting assumes an asterisk (*) and will give you a list of all the files in the default volume and directory. The default volume and directory names are displayed in the Executive Status frame. Since the default files list is an asterisk (*), you can get a listing of files by simply entering "Files" and pressing GO.

[Details?] Default: No

Enter "yes" if, for each file specified in the File list field, you wish to see information on file length, number of sectors, date and time last modified, and protection level.

[Print file] Default: No

If you wish to have the files list and/or details saved in a file, enter a file specification. If the file does not exist it will be created.

Note: If you enter the name of an existing file, the existing file will be deleted and the new file created in its stead.

If you enter the name of a printer, the files list will be sent to that printer.

Whether you choose the print file option or not, the files list will still be displayed on the screen.

FLOPPY COPY

Introduction

The Floppy Copy command allows you to copy disks using workstations with floppy disk drives.

Floppy Copy makes an exact duplicate of a disk. However, it does not duplicate disks that have unrecoverable "bad spots." If a master disk has bad spots, a new master disk without bad spots should be created (using the Backup Volume, IVolume, and Restore commands) before any duplication takes place.

This command is implemented using the Run file FloppyCopy.Run.

Command Form

Floppy Copy
[Number of copies] ____________________
[Overwrite ok?] ____________________
[Dual floppy?] ____________________
[Suppress verification?] ____________________
[Device name(s)] ____________________
[Device password(s)] ____________________

Parameter Fields

[Number of copies] Default: Unlimited

Enter the number of copies to be made from each master disk. This applies only to copies made using a single drive.

[Overwrite ok?] Default: No

If you enter "yes", the blank disk used for the copy is not checked to see if it is a valid volume.

[Dual floppy?]

Default: No for workstations with a hard disk

If you enter "yes", the copy is made with the master disk in floppy drive

[FØ] and the duplicate disk in floppy drive [F1].

If you enter "no", then one drive is used, with a temporary file on the Winchester disk holding the image of the master disk being duplicated.

[Suppress verification?] Default: No

If you enter "yes", then the program does not reread the master and duplicate disks to verify that no errors occurred. (Floppy Copy executes about 4Ø% faster if no verification is performed.)

[Device name(s)] Default: [FØ] [F1]

If you have renamed the [FØ] and [F1] then enter the new name(s).

The default names apply unless you renamed the disk drives when you built the operating system.

[Device password(s)] Default: No password

If you have your own version of the operating system with different passwords, specify them here.

FORMAT

Introduction

The Format command formats the text contained in one or more files into a paginated document that is printed by direct or spooled printing to a printer, or a specified file.

For spooled printing, the printer spooler creates a queue entry and appends it to the specified scheduling queue.

The defaults listed below produce text that fits an 8 1/2-by-11-inch page if the printer prints six vertical lines and ten horizontal spaces per inch.

For more information on the printing process, see the subsection "Printer Management" in section 5 "Advanced Concepts."

This command is implemented using the Run file Format.Run.

Command Form

Format
File list ____________
[Print to] ____________
[Confirm each?] ____________
[Title] ____________
[First page to format] ____________
[Last page to format] ____________
[Suppress page numbers?] ____________
[Suppress date?] ____________
[Suppress time?] ____________
[Double-space?] ____________
[Left margin (default 10 spaces)] ____________
[Text width (default 65 spaces)] ____________
[Page length (default 66 lines)] ____________
[Top margin (default 6 lines)] ____________
[Bottom margin (default 6 lines)] ____________
[Tab width (default 8 spaces)] ____________
[Suppress page ejects between files?] ____________

File list — Enter the file specification, file list, or at-file name for the file(s) you want to format.

[Print to] Default: [Spl] the spooled printer

Enter one of the following:

- The name of a local printer ([Ptr]A, [Ptr]B, or [Lpt]), enclosed in brackets, if the file is to be printed by direct printing.

- The name of a scheduling queue, enclosed in brackets, if the file is to be printed by spooled printing. The name must match a queue name defined for the system.

- The name of the file, not enclosed in brackets, into which the formatted image of the input file is written.

[Confirm each?] Default: No

If you enter "yes", you are prompted for confirmation before the file is formatted.

If you enter "no" or leave this field blank, Format proceeds without user interaction.

[Title] Default: Filename

Enter the title to be inserted at the top of each page of the file. If the title includes spaces, it must be enclosed in single quotes.

[First page to format] Default: 1

Enter the number of the first page to be formatted.

[Last page to format] Default: Last page

Enter the number of the last page to be formatted.

[Suppress page numbers?] Default: No

If you enter "yes", the page numbers

are not inserted at the top of each page of the file.

If you enter "no" or leave this field blank, the page numbers, are inserted at the top of each page of the file.

[Suppress date?]   Default: No

If you enter "yes", the date is not inserted at the top of each page of the file.

If you enter no" or leave this field blank, the date is inserted at the top of each page of the file.

[Suppress time?]   Default: No

If you enter "yes", the time is not inserted at the top of each page of the file.

If you enter "no" or leave this field blank, the time is inserted at the top of each page of the file.

[Double-space?]   Default: No

If you enter "yes", the lines are double-spaced.

If you enter "no" or leave the field blank, the lines are single-spaced.

[Left margin (default 10 spaces)]

Enter the number of spaces you want to leave blank for the left margin.

[Text width (default 65 spaces)]

Enter the number of spaces you want the text width of the file to occupy. Lines longer than specified are split into two or more lines. Split lines do not wrap into succeeding lines.

[Page length (default 66 lines)]

Enter the number of lines you want per page, including the top and bottom margins.

[Top margin (default 6 lines)]

Enter the number of blank lines you want to allow for the top margin.

[Bottom margin (default 6 lines)]

Enter the number of blank lines you want to allow for the bottom margin.

[Tab width (default 8 spaces)]

Enter the number of spaces you want between tab stops.

[Suppress page ejects between files?] Default: No

If you enter "yes", the files are printed as continuous text.

If you enter "no" or leave the field blank, a new page is started for each file.

Examples

To format three files into a single document and print it on the local printer on channel A, complete the format form as below.

```
Format
  File list                          File1 File2 File3
  [Print to]                         [Ptr]A
  [Confirm each?]
  [Title]                            'Working Draft 3'
  [First page to format]
  [Last page to format]
  [Suppress page numbers?]
  [Suppress date?]
  [Suppress time?]                   Yes
  [Double-space?]                    Yes
  [Left margin
      (default 10 spaces)]
  [Text width
      (default 65 spaces)]
  [Page length
      (default 66 lines)]
  [Top margin
      (default 6 lines)]
  [Bottom margin
      (default 6 lines)]
  [Tab width
      (default 8 spaces)]
  [Suppress page ejects
      between files?]                Yes
```

A single document composed of the three input files is printed double-spaced as continuous text. "Working Draft 3", the page number, and the date are printed at the top of each page of each file.

Introduction

The Install Batch command creates a secondary partition and installs a batch manager in it.

The Install Batch command is implemented with the Run file InstallBatch.Run.

This command is implemented using the Run file InstallBatch.Run.

Command Form

Install Batch
[Memory size (default = 2ØØK)] ________
[Batch queue (default = BATCH)] ________
[Partition name (default = BatchPartØØ)] ________
[Batch manager priority (default = 129)] ________

Parameter Fields

[Memory size (default = 2ØØK)]

Enter the memory size of the partition to be created. The size can be specified as number of kilobytes (1Ø24 bytes). The size should be larger than the memory requirement of the largest application that the batch manager will run.

[Batch queue (default = BATCH)]

Enter the name of the batch scheduling queue served by the batch manager. (See the "Batch Scheduling Queue" subsection in Section 7, "Batch Processing.")

[Partition name (default = BatchPartØØ)

Enter the name of the partition into which the batch manager is to be installed. If a partition with this name does not exist, one is created. If a partition with this name does exist, Install Batch attempts to install the batch manager in it.

If this field is defaulted and there is an already existing partition with the name BatchPart00, a partition is created with the name BatchPart01.

[Batch manager priority (default = 129)]

Enter the process priority (10-254, with 10 the highest) at which you want the batch manager to run. The batch manager assigns this same priority to the job step Run files. The default priority for the primary application partition is 129.

Introduction

The Install Queue Manager command installs the Queue Manager. The Queue Manager must be installed at the master workstation. See the subsection "Printer Management" in Section 5, "Advanced Concepts" and the section on queue management in the CTOS Operating System Manual for more information about the queue manager.

This command is implemented using the Run file InstallQMgr.Run.

Command Form

Install Queue Manager
[Use cache?] ______________________________

Parameter Fields

[Use cache?] Default: No

If you enter "yes", the installation of the Queue Manager issues an extra 4K-byte buffer. This buffer is used to optimize the performance of the Queue Manager.

If you enter "no" or leave this field blank, the Queue Manager uses less space, but will run slower.

Introduction

The Install Spooler command installs the spooler. The spooler can be installed in a master, cluster, or standalone workstation. (See the subsection "Printer Management" in Section 5, "Advanced Concepts" for more information on the spooler.)

This command is implemented using the Run file InstallSpl.Run.

Command Form

Install Spooler
  [Spooler configuration file] ________________

Parameter Fields

[Spooler configuration file]

Default: [Sys]<sys>SplCnfg.Sys

Enter the file specification for the spooler configuration file you want to use.

If you leave this field blank, the file [Sys]<sys>SplCnfg.Sys will be used.

IVOLUME

Introduction

The IVolume command prepares a medium such as a floppy or Winchester disk for use as a volume on your workstation.

IVolume formats the medium to accommodate 512-byte sectors, performs write/read tests to identify surface defects, writes volume control structures (such as the Volume Home Blocks, the Master File Directory, the Allocation Bit Map, and the File Header Blocks) onto the medium, and creates system files. (For more information on volume control structures, see the section on file management in the *CTOS Operating System Manual*.)

IVolume can

- verify the integrity of the medium
- identify bad sectors (which are not used for data storage)
- read the old list of bad sectors (when reinitializing a Convergent-formatted volume)
- recognize a Convergent-formatted volume and prompt for confirmation before deleting its files
- optimize storage space, number of files, etc.

Use of IVolume is discussed in detail below in the subsection "Description," which follows the "Parameter Fields" subsection.

This command is implemented using the Run file IVolume.Run.

Command Form

```
IVolume
  Device name                               _______________
   [Device password]                        _______________
  Volume name                               _______________
   [Volume password]                        _______________
   [System Image (default = 384)]           _______________
   [Log file (default = 2)]                 _______________
   [Crash file (default = Ø)]               _______________
   [Max. directories]                       _______________
   [Max. files on volume]                   _______________
   [Primary file headers only?]             _______________
   [Max. files in Sys Directory]            _______________
   [Sys Directory password]                 _______________
   [Write protect Sys Directory?]           _______________
   [Suppress format of medium?]             _______________
   [Surface tests]                          _______________
   [Debug?]                                 _______________
   [Log file]                               _______________
   [Extended floppy tracks?]                _______________
   [Single-sided mini-floppy?]              _______________
   [Bad spots]                              _______________
```

Parameter Fields

Device name

Enter the name of the device (assigned at system build) that contains the medium to be formatted.

For the version of the CTOS operating system on the distribution disk, the names of the floppy disk drives are FØ and F1 and the Winchester disk drives DØ, D1, and D2.

[Device password] Default: No password

Enter the password for the device that contains the medium to be initialized. The password is assigned originally at system build.

For the version of the CTOS operating system on the distribution disk, the device passwords of the Winchester disk drives are DØ, D1, and D2.

Volume name

Enter the name you want to assign to this volume. Volume names can have up to 12 characters. The name cannot duplicate any other volume or device name.

Invalid volume names are

- Comm_anything_
- CTOS_anything_
- DØ, D1, D2 FØ, F1
- Kbd
- Lpt_anything_
- Nul
- Spl_anything_
- Sys_anything_
- Tape_anything_
- Vid_anything_

[Volume password] Default: No password

Enter the password to be assigned to this volume. Passwords can have up to 12 characters. The password is used when creating directories or creating and opening files on this volume.

If no password is specified, the volume is unprotected. No directories or files can have passwords, and the volume can be overwritten at any time.

Volume protection can be specified later with the Change Volume Name command.

[System Image (default = 384)]

Enter the number of sectors required for the CTOS Run file. A sector is equal to 512 bytes. The Linker prints the size of the Run file at system build. (See the Linker/Librarian Manual.) It is not possible to expand the number of System Image sectors later, therefore, allow enough sectors for future expansion if there is a possibility of installing another CTOS Run file.

Note that 384 sectors are sufficient for all Convergent-generated System Images. Standalone or cluster System Images are smaller, so 256 sectors are sufficient for floppy-based systems.

IVolume creates the file <sys>SysImage.Sys but does not copy the System Image to it. That is done by the Copy command or the Restore command.

To initialize a volume that does not contain a CTOS System Image, specify 0. If 0 is specified, it is not possible to add a bootstrapable System Image file on this volume without reinitializing it.

[Log file (default = 2)]

Enter the number of Sectors required for the Log file. If the Log file tends to fill before it is convenient to print it, specify a larger number.

Since log entries are written only to system disks, if this volume does not contain a System Image, specify 0.

[Crash file (default = Ø)]

Enter the number of sectors required for the Crash Dump file. Upon bootstrapping after a system crash, the entire contents of memory are first dumped into the Crash Dump file, if it exists. This file can help determine the cause of the system crash.

Allocate two sectors for each 1K byte of memory to be dumped. If a number is not specified, a Crash Dump file is not written when the operating system is bootstrapped.

[Max. directories]

Enter the maximum number of directories that can be created on this volume. This number is actually rounded up to the nearest multiple of 15 to produce the size of Mfd.Sys. Since Mfd.Sys cannot be expanded, it is important to determine the absolute maximum.

The default depends on the type of device being initialized.

[Max. files on volume]

Default: Depends on device

Enter the maximum number of files that can be created on this volume. Allow a sufficient number for future expansion. It is not possible to expand this number at a later date without reinitializing the volume.

IVolume allocates File Header Block (FHB) sectors for each file. IVolume muliplies the number given by 1.5 to allow for a certain number of extension FHBs and rounds the number up to an even multiple of three times the number of sectors on each cylinder of the disk. It carefully locates these FHBs on the disk to ensure that the primary and secondary FHBs are located on different cylinders, at different rotational positions, and (for Winchester disks) on different surfaces.

The default depends on the type of device being initialized.

[Primary file headers only?] Default: No

If you enter "yes", IVolume allocates space for only a primary File Header Block for each file, rather than primary and secondary FHBs.

This option conserves disk space. However, operating system reliability is compromised, since an input/output error in its File Header Block makes a file unusable.

[Max. files in Sys Directory]

Default: Depends on device type

Enter the maximum number of files that can be created in the <Sys> Directory. This number is actually rounded up to the nearest multiple of 14. Since file system performance is degraded if this directory is more than 80% full, allow for more files than needed. In determining the maximum number of files, remember that the system files reside in the Sys Directory.

The default depends on the type of device being initialized.

[Sys Directory password] Default: No password

Enter the password (a string of up to 12 characters) to be assigned the <Sys> Directory. Leave this field blank if files in the <Sys> Directory are not to be password protected. If specified, this password must be presented whenever a file is created in the Directory. This password can also be used to authorize the opening of a file in the <Sys> Directory.

[Write protect Sys Directory?] Default: No

If you enter "yes", the default file protection level of the files in the <Sys> Directory is set to "modify protected." (A volume and Sys Directory password must have already been specified.)

[Suppress format of medium?] Default: No

If you enter "yes", the time to reinitialize a medium that was previously a Convergent-formatted volume is reduced.

Leave this field blank when initializing a new disk.

[Surface tests]

Default: Floppy disk 1
Initialized Winchester 1
New Hard disk 8

Enter the number of surface tests to be performed on each disk.

The surface test writes and reads each sector on the disk to ensure that it is defect free. Any errors are logged and an entry for that sector made in BadBlk.Sys. Random data is used for the surface tests. Therefore, the more passes specified, the more reliable the tests.

Surface tests should be run on floppy disks the first time they are initialized. Any floppy disks for which IVolume reports surface errors should be discarded.

Specifying Ø reduces the time to reinitialize a floppy disk.

The defaults for [Surface tests] are 1 for floppy disks, 1 for initialized Winchester disks, and 8 for uninitialized Winchester disks.

[Debug?]

Default: No

If you enter "yes", IVolume displays additional information that is only of interest to systems engineers.

[Log file]

Default: Screen display only

Enter the name of the file to which to write a report of the IVolume operation.

If the log file exists, the log is appended to it. If it does not exist, it is created.

If no log file is named, the log appears only on the screen.

[Extended floppy tracks] Default: No

If you enter "yes", the innermost six tracks of the floppy disks are used for storage.

If you enter "no", IVolume creates the file DiagTest.Sys to occupy these tracks. This file cannot be deleted, is not backed up, and should not be used to store data.

Using the extended (innermost six) tracks of floppy disks degrades floppy disk reliability but results in an 8% increase in the storage capacity of the floppy disk.

[Single-sided mini-floppy?] Default: No

If you enter "yes", only one side of the 5 1/4-inch mini-floppy is used.

If you enter "no", both sides of the 5 1/4-inch mini-floppy are used.

This field does not apply to an 8-inch floppy disk.

[Bad spots] Enter

t/h/sb/b

or

t/h/#sector

or

t/h/$sector

where, of the bad spots on the medium

t	Is the track number
h	Is the head number
sb	Is the starting byte number
bc	Is the number of bytes
#sector	Is the sector number of a 512K-byte sector
$sector	Is the sector number of a 256K-byte sector

The form *t/h* declares all sectors on track *t* and head *h* bad. For example, 3/Ø declares all sectors on track 3, head Ø, as bad.

The form *t/h/#sector*, used with IWS workstation disks, declares sector number *#sector* on track *t*, head *h*, as bad. For example, 3/Ø/#2 declares sector 2 on track 3, head Ø as bad. This form is compatible with the information displayed by the IWS Winchester disk diagnostic.

The form *t/h/$sector*, used with AWS workstation disks, declares sector number *$sector* on track *t*, head *h*, as bad. For example, 3/Ø/$4 declares sector 4 on track 3, head Ø as bad. This form is compatible with the information displayed by the AWS Winchester disk diagnostic and the "bad spot" report that is shipped with the mini-Winchester drive.

All numbers specified are decimal. To specify a hexadecimal number, append the letter "h" to the end of the number. For example:

```
3/Ø/ØCh
```

is the same as

```
3/Ø/12
```

Separate the list of bad spots with blanks, for example:

```
3/Ø/1Ø1/26 7/1 16/2/#2
```

The form accommodates up to three lines of bad spot information. The description of each bad spot must be contained on a single line.

IVolume performs a surface test to identify bad spots on the medium. However, there are certain spots that are marginal and cannot be found by the surface test. Marginally bad spots do not always fail, but they can, and are thus not appropriate areas in which to store data.

A report of bad spots is shipped with each Winchester disk drive. This report includes all bad spots at the time of factory test, including marginal ones. Enter this list manually whenever a new disk is initialized.

When IVolume is used to reinitialize a disk, it reads the old BadBlk.Sys file to find the current list of bad spots.

If its surface test identifies additional bad spots, IVolume adds these to the list of bad spots in BadBlk.Sys.

The Winchester or floppy disk diagnostic destroys the contents of all files. On the next use of IVolume, reenter the list of bad spots.

Description

Protection. Since IVolume can destroy data, it is advisable to protect it from unauthorized use. Use the Set Protection command to protect the run file (IVolume.Run).

> CAUTION
>
> Because areas of a disk surface can be marginal, it is critical to manually specify the list of bad spots identified by the report shipped with your Winchester disk.

Operation. IVolume verifies the consistency of the specified parameters, opens the log file (if one is specified), and displays the values chosen for the sizes of the volume control structures.

When IVolume is initializing a floppy disk, it prompts you to mount the floppy disk.

IVolume does not allow initializing a volume currently in use (the System Volume [Sys], volumes in use by other cluster workstations, etc).

To avoid inadvertently destroying a volume that contains valuable data, IVolume reads the sectors in which a Volume Home Block(VHB) exists on a Convergent-formatted volume. If a valid VHB is found, IVolume displays the information contained in it, and asks for confirmation of the reinitialization of the volume. This message is displayed, with the information filled in:

```
Medium contains valid volume information.
  Volume name:
  Creation date/time:
  Last modification date/time:
  Number of free sectors:
  Number of free file headers:

Do you wish to ERASE this volume?  (Press GO
to confirm, CANCEL to deny, or FINISH to
return to the Executive.)
```

If you press GO, and the volume has a password, IVolume requires the old password to be supplied as added confirmation, with the prompt:

```
Password?
```

IVolume echoes a pound sign (#) for each character typed as the password is entered. Terminate the password by pressing RETURN.

IVolume formats the medium and performs the number of surface tests specified. Each bad spot is recorded in the bad sector file (BadBlk.Sys). At the end, the list of bad sectors is displayed.

Sectors are identified in the bad sector file as defective (and omitted from the Allocation Bit Map) for three reasons:

1. The medium contains a valid Convergent volume and the sector was previously identified in the Bad Sector File as defective

2. The surface test identified the sector as defective

3. The sector was specified in [Bad spots] of the IVolume form. (Because areas of a disk

> surface can be marginal, it is critical to manually specify the list of bad spots identified by the report shipped with your Winchester disk.)

Volume control structures (the files Mfd.Sys, FileHeaders.Sys, BadBlk.Sys, DiagTest.Sys, the VHBs, and Allocation Bit Map) are written on the medium, and the system files SysImage.Sys and CrashDump.Sys are created. IVolume does not, however, move any data to these files. A CTOS System Image must be copied to SysImage.Sys using the Copy command or the Restore command. System files can only be created by IVolume. For example, SysImage.Sys cannot be copied to an initialized volume without space being allocated for it by IVolume.

After IVolume initializes the medium, if it was a floppy disk, IVolume prompts you with

> Initialize another volume? (Press GO to confirm, CANCEL to deny, or FINISH to return to the Executive.)

This is useful for preparing a number of floppy disks for backup, all with the same volume name. To initialize another floppy disk, insert it in the drive and press GO. It is initialized with the same parameters as the previous one.

DiagTest.Sys file. The DiagTest.Sys file is reserved for diagnostics. It cannot be deleted, is not backed up, and should not be used to store data.

The DiagTest.Sys file for a Winchester disk is allocated on the last track of the disk.

The DiagTest.Sys file for a floppy disk is allocated on the innermost six tracks if you enter "no" in the [Extended floppy tracks?] field.

Optimal IVolume Parameters. Table 6-3 below shows the optimal IVolume parameters for a system Winchester disk, a nonsystem Winchester disk, a system floppy disk, a nonsystem floppy disk, and an archive floppy disk. The device names and passwords given here reflect those used in the standard release version of the CTOS Operating System. They may differ, however, if an installation has performed a customized system build. Volume names and passwords can be chosen independently. If a Winchester disk is being initialized for the first time (or after performing a Winchester disk diagnostic), fill in [Bad spots] with the list of bad spots; otherwise, leave the field blank.

For a System Volume (that is, a volume that contains the CTOS System Image [Sys]<Sys>SysImage.Sys), only the device name and password (if one exists), and the volume name must be specified.

All numbers in the form are decimal.

Table 6-3. Optimal IVolume Parameters.

IVolume	System Winchester Disk	Winchester Disk on Mass Storage Expansion	System Floppy Disk	File Floppy Disk	Archive Floppy Disk
Device name	D0	D1	F0	F0	F0
[Device password]	D0	D1	F0	F0	F0
Volume name	Win	Win1	Floppy	Files	Archive
[Volume password]	Win	Win1			
[System Image (default = 384)]		0	†	0	0
[Log file (default = 2)]		0		0	0
[Crash file (default = 0)]					
[Max. directories]					1
[Max. files on volume]					10
[Primary file headers only?]					Y
[Max. files in Sys Directory]					10
[Sys Directory password]					
[Write protect Sys directory?]					
[Suppress format of medium?]					
[Surface tests]					
[Debug?]					
[Log file]					
[Extended floppy tracks?]					
[Single-sided mini-floppy?]					
[Bad spots]					

† 256 is sufficient for standalone operating systems.

LCOPY

Introduction

The LCopy command copies one or more files. LCopy allows you to split up the file specification into three parts, a prefix, name, and suffix. (Prefix here is used in the ordinary sense and does not refer to a "file prefix.") The prefix, name, and suffix are combined logically to form a file specification before the file is copied. If any of these parts are not specified, a null string is used instead.

If you wish to copy a file or list of files to a new volume or a new directory without renaming the file, you can do this more easily with LCopy than with the Copy command. LCopy is especially useful for copying all the files used for an application system onto a floppy, etc.

Passwords can be specified for the source and destination files using the carat (^). (See the subsection "File Specification" in Section 4 "Basic Concepts.") Null passwords are ignored. Passwords longer than 12 characters cannot be used and cause LCopy to terminate (with status code 217). If the specified password is invalid, the logged-in password is used.

The LCopy command differs from the Copy command in that it copies actual files rather than using device byte streams. (See the section on sequential access methods in the *CTOS Operating System Manual*.) Also, it has more sophisticated argument processing.

This command is implemented using the Run file LCopy.Run.

Command Form

LCopy

File list ________________________

[File prefix(es) from] ________________________

[File prefix(es) to] ________________________

[File suffix(es)] ________________________

[Overwrite ok?] ________________________

[Confirm each?] ________________________

[Continue on error?] ________________________

[Verify copy?] ________________________

Parameter Fields

File list

Enter the name(s) of the file(s) you want to copy. You can use the wild card characters (* or ?) in this field, or an at-file.

If you include the node, volume, and directory names here, they are removed from the file name for creating the source and destination file specifications.

[File prefix(es) from] Default: Null string

Enter the string of characters you want prefixed to the file name entered in the File list field to construct the source file specification. The number of entries in this field must be equal to the number of entries in the [File prefix(es) to] field. Entries are paired on a one-to-one basis.

The node, volume, and directory names are removed from the file name of entries made in this field before the prefixes are added.

Passwords specified here, using the carat (^)as a part of the file specification, apply to the source file if no password is specified in the File list field. (See the "File Protection" and "Passwords" subsections in Section 4, "Basic Concepts.")

[File prefix(es) to] Default: Null string

Enter the string of characters you want prefixed to the file name entered in the File list field to construct the destination file specification. The number of entries in this field must be equal to the number of entries in the [File prefix(es) from] field. Entries are paired on a one-to-one basis.

If one or no entry is made in the [File prefix(es) from] field, then the source file is copied to each of the file names constructed using the entries of this field.

Passwords specified here, using the carat (^) as a part of the file specification, apply to the destination file if no password is specified in the [File suffix(es)] field.

[File suffix(es)]   Default: Null string

Enter the string of characters you want used as a suffix for the source and destination file specification. If there is more than one entry in this field, file specifications using each entry are constructed.

Passwords specified here, using the carat (^) as a part of the file specification, apply to the destination file if no password is specified in the [File prefix(es) to] field.

[Overwrite ok?]   Default: Ask for Confirmation

If you enter "yes" and the destination file already exists, the existing file is deleted and the LCopy operation continues.

If you enter "no" and the destination file already exists, that file is not copied.

If you leave the field blank, the Executive will prompt you for confirmation before the existing file is deleted. You can confirm this by pressing GO, deny it by pressing CANCEL, or stop copying files by pressing FINISH.

[Confirm each?]   Default: No

If you enter "yes", then you are prompted to confirm each LCopy by pressing GO, deny it by pressing CANCEL, or stop copying files by pressing FINISH.

If you enter "no" or leave the field blank, then you are not prompted for confirmation.

[Continue on error?] Default: Ask for Confirmation

If you enter "yes", errors are reported and LCopy continues.

If you enter "no", LCopy terminates when an error is encountered.

If you leave the field blank, LCopy stops and asks for confirmation to continue.

[Verify copy?] Default: No verification

If you enter "yes", LCopy compares the source and destination files during the copy.

Examples

The form below copies everything from the <CT> directory of a floppy disk to [Sys]<Sys>:

LCopy

File list	[FØ]<CT>*
[File prefix(es) from]	
[File prefix(es) to]	[Sys]<Sys>
[File suffix(es)]	
[Overwrite ok?]	Yes
[Confirm each?]	
[Continue on error?]	
[Verify copy?]	

The form below copies the payroll files from the hard disk to a floppy:

LCopy

File list	Payroll.*
[File prefix(es) from]	[DØ]<Accounting>
[File prefix(es) to]	[FØ]<Accounting>
[File suffix(es)]	
[Overwrite ok?]	y
[Confirm each?]	
[Continue on error?]	
[Verify copy?]	y

The form below copies CtLfs.Run from a floppy to SysImage.Sys (overwriting the existing file), and then verifies that the destination file was copied correctly. In this example LCopy is being used as you would use Copy, but verification is performed:

```
LCopy
  File list
  [File prefix(s) from]      [FØ]<CT>CtLfs.Run
  [File prefix(s) to]        [Sys]<Sys>SysImage.Sys
  [File suffix(s)]
  [Overwrite ok?]            Yes
  [Confirm each?]
  [Continue on error?]
  [Verify copy?]             Yes
```

LOGOUT

Introduction

The Logout command terminates the current user session, causes the workstation to reset any information previously specified with the SignOn form or with the Path command, and reinitializes the screen.

This command is implemented using the Run file SignOn.Run.

Command Form

Logout

Parameter Fields

This command has no fields.

Introduction

The Maintain File command modifies and/or reads data files. These files can be Record Sequential Access Method (RSAM) files, Direct Access Method (DAM) files, or the data store files of Indexed Sequential Access Method (ISAM) data sets. (RSAM and DAM are described in the sections of those names in the CTOS Operating System Manual, and ISAM is described in the ISAM Manual.)

Maintain File can

- Verify the file structure.

- Remove malformed records.

- Remove deleted records.

- Optionally write a log of the verification of the file structure to a file. (The log always appears on the screen.)

This command is implemented using the Run file MaintainFile.Run.

Command Form

Maintain File
  Input files ____________________
  [Output file] ____________________
  [Log file] ____________________
  [Remove deleted records?] ____________________
  [Suppress confirmation?] ____________________

Parameter Fields

Input files

Enter a list of the names of one or more data files (RSAM, DAM, or ISAM data store) to be maintained. If this is the only field filled in, the file structure is verified, and a report of the verification appears on the screen. (The contents of this report are described below under the [Log file] field.)

[Output file] Default: No new file created

Enter the name of the file to which to copy well-formed input records.

If all the input files are readable using DAM, the output file is a DAM file; otherwise, the output file is an RSAM file.

If the file specification you enter is for a file that already exists, its existing contents are discarded. If it does not exist, it is created.

The treatment of deleted and malformed records in the output file is discussed under [Remove deleted records?] below.

[Log file] Default: Log file not saved

Enter the name of the Log file to which to write a report of the verification of the file structure. The Log file lists

- o the number of records processed
- o the number of bytes of data processed
- o the logical file address and length of any malformed records

If no Log file is named, the report appears only on the screen.

[Remove deleted records?] Default: No

Enter "yes" or "no". The treatment of deleted (and malformed) records depends on whether the output file is an RSAM or a DAM file.

If the output file is an RSAM file, deleted and malformed records are always skipped.

If the output file is a DAM file, and you entered "no", then deleted records in the input file cause corresponding deleted records to be created in the same positions in the output file. Malformed records are also treated as deleted records in the output file.

If the output file is a DAM file, and you enter "yes", deleted and malformed records are skipped and no corresponding ones created in the output file.

[Suppress confirmation?] Default: No

Enter "yes" or "no". When a malformed record is found in the input file, a message appears on the screen that specifies its logical file address and its length.

If you enter "no", Maintain file asks the user how to proceed. The options are: (1) to skip the malformed record and keep processing the input file, or (2) to terminate processing.

If you enter "yes", Maintain file proceeds without user interaction.

Example

The form below merges the two ISAM data sets <Jones>Order Line.Isam and <Smith>OrderLine.Isam into the ISAM data set <Mstr>Temp, and writes a log of the verification report:

```
Maintain File
  Input files                   <Jones>OrderLine.Isam
                                <Smith>OrderLine.Isam
  [Output file]                 <Mstr>Temp
  [Log file]                    <Mstr>Temp.Log
  [Remove deleted records?]     ____________________
  [Suppress confirmation?]      ____________________
```

Maintain file scans the data store files of the two data sets, verifying the file structures and reclaiming the space occupied by deleted records, and merges the well-formed records into <Mstr>Temp. Figure 6-9 below shows a sample Log file.

```
Malformed input record in input file Ø at Lfa 512

189 bytes skipped before a properly formed record was
found.

45 records read. 2835 bytes read.
```

Figure 6-9. Sample Log File.

Status Code

Hexadecimal Value	Decimal Value	Meaning
11ØØ	Ø44C	Malformed input record. A malformed input record was found and Maintain file was terminated.

Introduction

The New Command command adds a command to those recognized by the Executive.

To understand how the information in a New Command command is passed to the operating system, see the section on parameter management in the CTOS Operating System Manual.

For an example of how to implement a new command see the subsection "Adding a New Command" in Section 5, "Advanced Concepts."

This command is implemented using the Run file Command.Run.

Command Form

New Command

Command name ______________________

Run file ______________________

Field names ______________________

Description ______________________

[Overwrite ok?] ______________________

[Case (default 'ØØ')] ______________________

[Command file] ______________________

Parameter Fields

Command name

Enter the name you wish to assign to the command. This command name will be used to invoke the run file named in the Run file field. Command names can include more than one word. If you use more than one word for the command name enclose it in single quotes (').

Run file

Enter the file specification for the Run file that is invoked by this command. The Run file must have been compiled or interpreted and linked.

Some commands are implemented within the Executive program, rather than with a separate Run file. For a discussion of this see the subsection "Adding a New Command" in Section 5. "Advanced Concepts."

Field names

Enter the parameter prompts for each line in the new command form. Each parameter will label one command line.

If the parameter prompts will not fit in this parameter field, create an at-file containing the parameter prompts you want to use.

You can control the way each field of the new command expands the wild card character. By default, the wild card character is expanded when you press RETURN. To suppress expansion of the wild card character until after the GO key is pressed, enter the simple parameter describing that command line followed by an asterisk (*). For an example see below.

Description

Enter the description of the command to be displayed by HELP.

[Overwrite ok?] Default: Ask for Confirmation

If you enter "yes" and the command named already exists, then the old command is removed from the Command file and is replaced by the new command.

If you enter "no" and the command named already exists, the new command will be cancelled.

If you leave the field blank, the Executive will prompt you for confirmation before the existing command is overwritten. You can confirm this by pressing GO, deny it by pressing CANCEL, or stop the new command operation by pressing FINISH.

[Case (default 'ØØ')]

If the Run file can be invoked by more than one command, enter the case value for this command.

Use of case value is discussed in the subsection "Adding a New Command" in Section 5, "Advanced Concepts."

Case may be any two characters. This field is always interpreted as a character string.

[Command file]

Default: Current Exec Command File

Enter a Command file name different from the one currently in use.

Description

Special Use of the Wild Card Characters. Some commands, such as Copy and Rename, give special treatment to the wild card characters, the asterisk (*) and the question mark (?). For most commands, when you press RETURN after filling in a field of the command form, any parameters that include a wild card character are expanded to the list of matching file names. Such commands are supported with a feature that causes the Executive to suppress the wild card expansion when you press RETURN.

If the last character of a field name specified to New Command is an asterisk (*), then the expansion of wild card characters is suppressed for parameters that will be typed in that field. In addition, the asterisk (*) in the field name is not displayed. That is, it does not appear in the command form you are creating; however, it remains known to the Executive.

For example, the form below shows how you complete the New Command command form to add the Copy command:

```
New Command
  Command name              Copy________________________
  Run file                  !2__________________________
  Field names               'File from*' 'File to*'_____
                            '[Overwrite ok?]____________

  Description               'Copy a file to another_____
                            file.'______________________

  [Overwrite ok?]           ____________________________
  [Case (default 'ØØ')]     ____________________________
  [Command file]            ____________________________
```

The resulting command form for Copy is:

```
  Copy
    File from            ____________________________
    File to              ____________________________
    [Overwrite ok?]      ____________________________
    [Confirm each?]      ____________________________
```

An asterisk (*) or a question mark (?) typed in a file specification to either the File from or File to fields is not expanded when you press RETURN. Rather, the Copy command expands it after you press GO.

When the Type command was created with the New Command command, an asterisk (*) was not used in the File list field. Therefore, if you use a wild card character in the File list field and press RETURN, the file specification is expanded in place.

Password Fields. A command form can include a password field, which does not display the password on the screen. Instead, the characters typed appear as pound signs (#).

A field is a password field if the last character in the field name specified to the New Command command is a carat (^). (The carat does not appear in the command form you are creating; however, it remains known to the Executive.)

For example, the form below shows how you complete the New Command command form to add the Path command:

```
New Command
  Command name              Path
  Run file                  !7
  Field names               'Volume' 'Directory'
                            '[Default file prefix]'
                            '[password]^' '[Node]'
  Description               'Set the ...'
  [Overwrite ok?]
  [Case (default 'ØØ')]
  [Command file]
```

The fourth field, [password]^, is a password field.

Introduction

The Partition Status command performs the following operations through its subcommands:

- displays status information about a specified application partition (Display Partition Status)
- loads a primary task in a specified application partition (Load Primary Task)
- terminates and vacates a specified application partition (Vacate Partition)
- removes the specified vacant application partition (Remove Partition)

When Partition Status is invoked, the Main display appears on the screen. This display is used to specify a partition and to invoke the subcommands (see "Subcommands" below) that perform the operations listed above. Figure 6-10 shows an example of a Main display.

The Partition Status command has four subcommands

- Display Partition Status
- Load Primary Task
- Remove Partition
- Vacate Partition

Each subcommand is described below.

This command is implemented using the Run file PartitionStatus.Run.

Command Form

The Partition Status command has no parameters.

Description

Main Display. Figure 6-10 shows an example of a Main display and lists three application partitions: one active and two vacant. The active partition, named "Primary", is 235K in size and the Run file currently executing in it is [Sys]<Sys>PartitionStatus.Run. The two vacant partitions, "Background00" and "Background01", are 31K and 41K in size, respectively. There is no Run file executing in either vacant application partition.

To invoke any of the subcommands shown at the bottom of the display, the partition named "Primary" is selected, and then the desired subcommand is selected. (See the subsection "Invoking Subcommands" below.)

Modes. Partition Status operates in either of two modes: operational or monitor. The current mode is shown in the upper left corner of the Main display. When Partition Status is invoked, operational mode is active. To switch to monitor mode, use the TAB, right arrow, or left arrow key to select "Monitor Mode" in the Main display, and press GO; or type the letter "X". Modes can be switched at any time.

Operational Mode. Partition Status must be in operational mode to invoke any of the subcommands.

The Main display is not automatically updated in operational mode. However, you can manually update the Main display with the PREV PAGE key. If there is more than one page of entries, use NEXT PAGE to page through the succeeding ones.

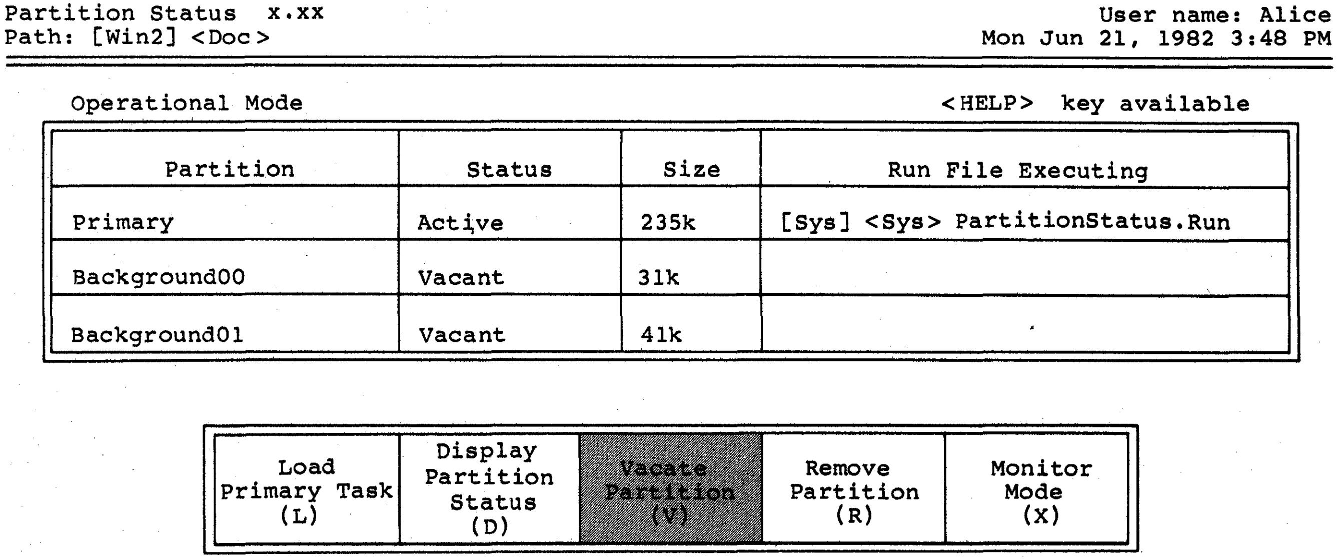
Monitor Mode. Monitor mode is used to monitor the progress of the run file currently executing in the specified application partition. During monitor mode, the Main display is updated every 30 seconds.

Invoking Subcommands

To invoke a subcommand, select an application partition name on the Main display on which to operate.

The selected application partition name is indicated by a movable highlight. Use the NEXT, RETURN, up arrow, or down arrow key to change the selected application partition name.

After selecting the application partition name, there are two ways to invoke the subcommand: type the single letter located below the subcommand name, or use the TAB, right arrow, or left arrow key to move the highlight to the desired subcommand and press GO. For example, to invoke the Vacate Partition subcommand, either type the letter V, or move the highlight to the box stating "Vacate Partition" and press GO.



```
Partition Status  x.xx                                    User name: Alice
Path: [Win2] <Doc>                                Mon Jun 21, 1982 3:48 PM
```

Operational Mode <HELP> key available

Partition	Status	Size	Run File Executing
Primary	Active	235k	[Sys] <Sys> PartitionStatus.Run
Background00	Vacant	31k	
Background01	Vacant	41k	

Figure 6-1Ø. Partition Status Main Display.

Display Partition Status

The Display Partition Status subcommand displays status information for the specified application partition, including

- its size
- its memory boundaries
- the name of the run file currently executing in it
- the termination status code and termination message from a previous task

There are two types of application partitions: active and vacant.

Subcommand Form

Display Partition Status
Partition name ______________________________

Subcommand Parameter Fields

Partition name

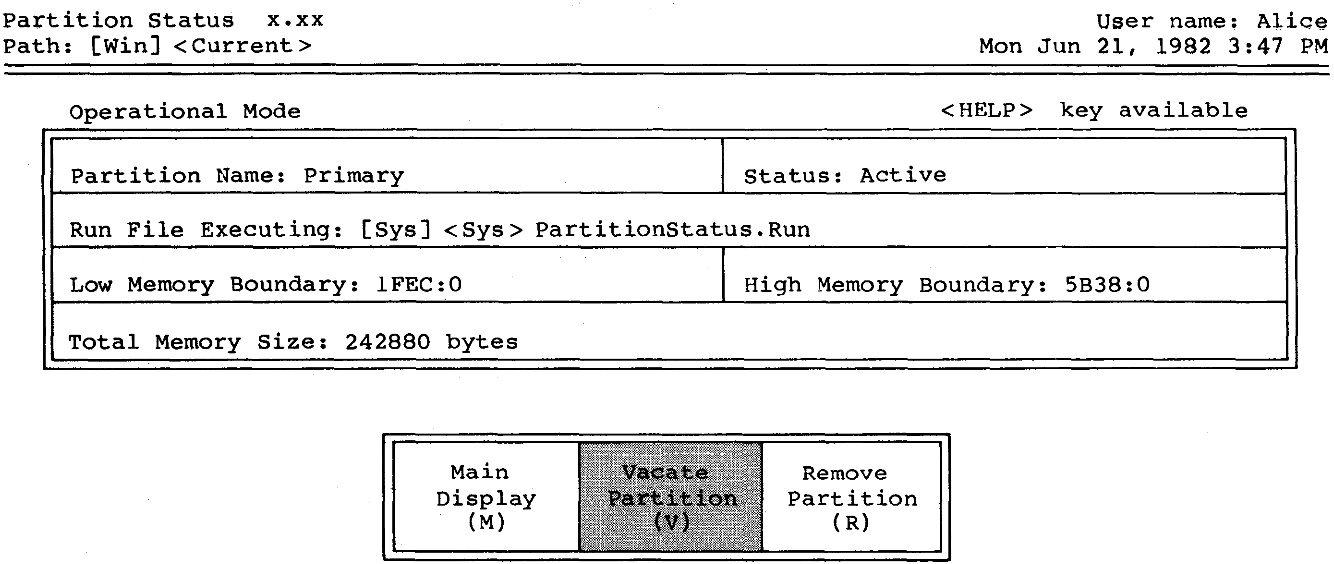
Enter the name of the specified application partition (active or vacant, as appropriate).

Active Partition Status Display. Figure 6-11 is an example of an Active Partition Status display and shows the status of the application partition named "Primary." Its active status and run file are shown. "Low Memory Boundary" is 1FEC:Ø, "High Memory Boundary" is 5B38:Ø, and "Total Memory Size" is 24288Ø bytes.

You can return to the Main display or invoke the Vacate Partition and Remove Partition subcommands (see their descriptions below) from the Active Partition Status display.

Vacant Partition Status Display. Figure 6-12 shows an example of a Vacant Partition Status display and shows the status of the selected vacant application partition. No previous run file is shown. "Low Memory Boundary" is 65A5:Ø and "High Memory Boundary" is 6D92:Ø. "Total Memory Size" is 32464 bytes. No termination status code or termination message is shown.

You can return to the Main display or invoke the Load Primary Task and Remove Partition subcommands (see their descriptions below) from the Vacant Partition Status display.



Partition Status x.xx
Path: [Win] <Current>

User name: Alice
Mon Jun 21, 1982 3:47 PM

Operational Mode

<HELP> key available

Partition Name: Primary

Status: Active

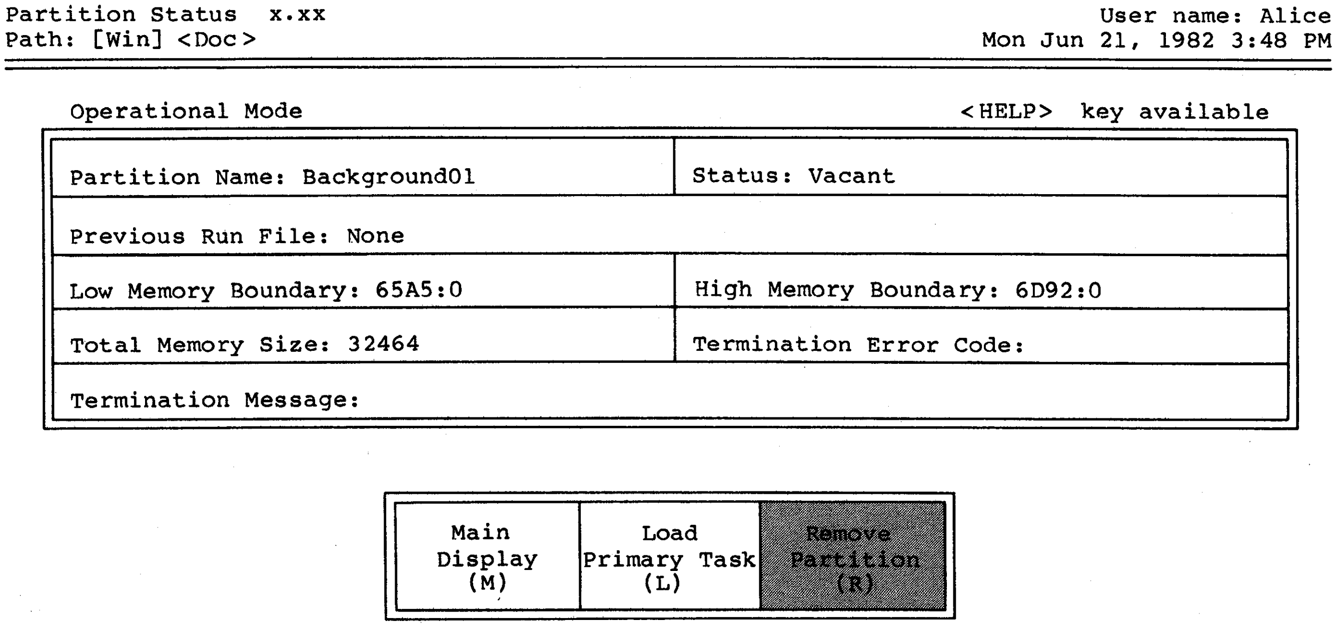
Run File Executing: [Sys] <Sys> PartitionStatus.Run

Low Memory Boundary: 1FEC:0

High Memory Boundary: 5B38:0

Total Memory Size: 242880 bytes

Figure 6-11. Active Partition Status Display.



Partition Status x.xx
Path: [Win] <Doc>

User name: Alice
Mon Jun 21, 1982 3:48 PM

Operational Mode

<HELP> key available

Partition Name: Background01

Status: Vacant

Previous Run File: None

Low Memory Boundary: 65A5:0

High Memory Boundary: 6D92:0

Total Memory Size: 32464

Termination Error Code:

Termination Message:

Figure 6-12. Vacant Partition Status Display.

Load Primary Task

The Load Primary Task subcommand loads the primary task into the specified application partition.

Subcommand Form

Load Primary Task
Partition name ________________________
Run file ________________________
[Priority] ________________________

Subcommand Parameter Fields

Partition name

Enter the name of the application partition.

Run file — Enter the full file specification of the primary task.

[Priority] — Default:129

Enter the process priority (1Ø-254, where 1Ø is the highest priority) of the primary task.

Remove Partition

The Remove Partition subcommand removes the specified vacant application partition.

Subcommand Form

Remove Partition
Partition name ________________________________

Subcommand Parameter Fields

Partition name

Enter the name of the application partition you want to remove.

Vacate Partition

The Vacate Partition subcommand terminates all tasks in the specified application partition but does not load and activate the partition's exit run file. The partition becomes vacant.

Subcommand Form

Vacate Partition
  Partition name ________________________________

Subcommand Parameter Fields

Partition name

Enter the name of the application partition in which to terminate tasks and vacate.

PATH

Introduction

The Path command sets default values that are used for node name, volume name, directory name, file prefix, and password when they are not entered with the file specification.

All the fields in this command are optional, so that you need only fill in the one you specifically wish to change.

This command is implemented internally by the Executive using the value !7.

Command Form

Path
[Volume] ____________________
[Directory] ____________________
[Prefix] ____________________
[Password] ____________________
[Node] ____________________

Parameter Fields

[Volume] Default: Logged in volume

Enter an existing volume name. Brackets [] are not necessary.

[Directory] Default: Logged in directory

Enter an existing directory name. Brackets <> are not necessary.

[Prefix] Default: No change to file prefix

Enter a file prefix. This character string is prefixed to any subsequent file specification you give.

This prefix will remain until you change it or remove it by entering a null string ('').

[Password] Default: No password

Enter a volume, directory, or file password. The password entered here will be used every time you specify a file until you enter a new password or remove the entry from this command by entering a null string ('').

[Node] Default: Logged in node

Enter an existing node name. Brackets ({}) are not necessary.

Introduction

The PLog command lists the content of the Log file. PLog can optionally write the log to a file or printer.

This command is implemented using the Run file Plog.Run.

Command Form

PLog
[Error type (Cr,B,In,D,Cl,Is)] ________________
[Print to] ________________
[Volume name] ________________
[After date/time] ________________

Parameter Fields

Error type (Cr,B,In,D,Cl,Is)] Default: All Types

Enter the type of errors you wish to display from the Log file. The six error types you can enter are

- o B = system bootstrap
- o Cr = system termination
- o Cl = cluster communications
- o D = disk
- o In = system initialization
- o Is = ISAM

The default is to show all errors.

[Print to] Default: Screen

Enter the name of the file or printer to which the Log file should be written. The requested information from the Log file is always written to the screen.

[Volume name]   Default: Current System Volume

Enter the name of the volume for which to list the content of the Log file. A Log file is maintained only for system volumes and only if specified in the IVolume command.

The default is to list the content of the Log file for the current system volume.

[After date/time]   Default: Current date and time

Enter the starting date/time for error entries to be printed. The date/time is entered as in the Set Time command or in the SignOn form.

Description

Log File. The Log file ([Sys]<Sys>Log.Sys) is an error- logging file. An entry is placed in it for the following types of system errors and events:

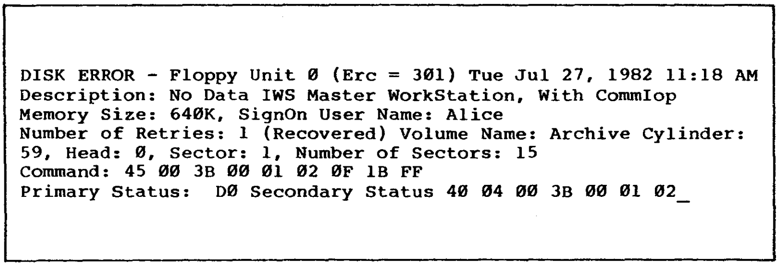
- system termination
- system bootstrap
- system initialization
- disk
- cluster communications
- ISAM.

Each entry in the Log file contains the date and time of the error, the error type, and additional information about the error. As the Log file fills, newer entries replace older ones. The file should therefore be inspected periodically to identify marginally functioning hardware or ISAM data base errors.

Status code 29Ø ("Log buffer overflow") indicates that multiple errors occurred rapidly and the operating system was unable to log all of them.

Example

Figure 6-13 shows a sample Log file.



```
DISK ERROR - Floppy Unit Ø (Erc = 3Ø1) Tue Jul 27, 1982 11:18 AM
Description: No Data IWS Master WorkStation, With CommIop
Memory Size: 64ØK, SignOn User Name: Alice
Number of Retries: 1 (Recovered) Volume Name: Archive Cylinder:
59, Head: Ø, Sector: 1, Number of Sectors: 15
Command: 45 ØØ 3B ØØ Ø1 Ø2 ØF 1B FF
Primary Status:  DØ Secondary Status 4Ø Ø4 ØØ 3B ØØ Ø1 Ø2_
```

Figure 6-13. PLog Sample Log File.

PRINT

Introduction

The Print command adds a file to the scheduling queue for spooled printing. Print does not create a temporary file as Format does. Rather, the actual file is queued for spooled printing. Therefore, it must not be deleted or modified until its printing is complete.

Command Form

```
Print
  File list                      ____________________
  [Queue name]                   ____________________
  [Delete after printing?]       ____________________
  [Number of copies]             ____________________
  [Special forms name]           ____________________
  [Print wheel name]             ____________________
  [Priority]                     ____________________
  [Printing mode]                ____________________
  [Align form?]                  ____________________
  [After date time]              ____________________
  [Security mode?]               ____________________
  [Confirm each?]                ____________________
```

Parameter Fields

File list

Enter the name(s) of the file(s) you wish to print.

[Queue name] Default: Spl

Enter the name of the scheduling queue to contain the queue entry generated by this request for printing. The name must match the queue name defined for the system.

[Delete after printing?] Default: No

If you enter "yes", the file is deleted.

If you enter "no" or leave the field blank, the file is not deleted.

[Number of copies] Default: 1 copy

Enter the number of copies you want printed.

[Special forms name] Default: No name

If you want to print the file on special paper, enter the name for it. This name can have up to 12 characters. This causes the printer to pause before printing so that the special paper can be loaded.

If you enter "no" or leave the field blank, the printer does not pause.

[Print wheel name] Default: No wheel specified

Enter the name of the print wheel you wish to use to print the file. This name can have up to 12 characters. If a name is specified, the printer pauses before printing so that the the print wheel can be changed.

To restart the printer, invoke the Spooler Status command and use the Restart printer subcommand.

[Priority] Default: 5

Enter the priority number you want to assign to the file in the scheduling queue. Priorities are Ø-9, with Ø as the highest.

[Printing mode] Default: N

Enter "N" for normal mode, "I" for image mode, or "B" for binary mode. See the subsection "Printer Management" in Section 5, "Advanced Concepts," for more information on printing mode.

[Align form?] Default: No

If you enter "yes", the printer pauses after the first page is printed so that the paper can be aligned. To restart the printer, invoke the

Spooler Status command and use the Restart Printer subcommand.

If you enter "no" or leave the field blank the printer does not pause for forms alignment.

[After date time] Default: ASAP

Enter the date and time after which you want the file to be printed. The format is

```
Mon Jun 1, 1983 8:ØØ pm
```

If you leave this field blank, the file will be printed as soon as possible.

[Security mode?] Default: No

If you enter "yes", the printer pauses before the file is printed. The user must enter a password from the workstation where the printer is connected before the file can be printed. Passwords are entered with the Enter Password subcommand of the Spooler Status command.

[Confirm each?] Default: No

If you enter "yes", you are prompted for confirmation before each file is printed.

If you enter "no" or leave the field blank, each file specified in the File list field is printed without a prompt for confirmation.

Introduction

The Record command records all subsequent keystrokes in a file until you give the Stop Record command. This file can later be used as a Submit file.

Note that a Submit file that works properly when the screen is in 132-column mode may not work if the screen is in 8Ø-column mode. This is because parameters entered for Executive forms in 132-column mode may not fit in 8Ø-column mode.

See the subsection "Using a Submit File" in Section 5, "Advanced Concepts," for a detailed explanation.

This command is implemented internally by the Executive using the value !19.

Command Form

Record
  File to record on ______________________________

Parameter Fields

File to record on

Enter the name you want to assign to the file.

Introduction

The Remove Command command takes the specified command name off the list of commands recognized by the Executive. The Executive will no longer recognize the command or display a HELP description for it.

This command is implemented using the Run file Command.Run.

Command Form

Remove Command
Old command name ____________________
[Command file] ____________________

Parameter Fields

Old command name

Enter the command name you want to remove. Abbreviations of command names are not allowed. You must enclose the command name in single quotes ('...') if it includes spaces.

[Command file]

Default: Current Executive Command file

Enter the file specification for a Command file not currently in use by the Executive. Example: If different Command files are associated with different users of the same workstation, you could use this field to remove a command for another user without having to log in using that user name.

REMOVE DIRECTORY

Introduction

The Remove Directory command removes a directory from a volume. The command optionally deletes all the files in the directory as well.

This command is implemented internally by the Executive using the value !15.

Command Form

Remove Directory
Old directory name ________________
[Volume or directory password] ________________
[Delete all files?] ________________
[Confirm while deleting?] ________________

Parameter Fields

Old directory name

Enter the name of the directory you wish to delete.

[Volume or directory password] Default: None

If there is a directory password, enter it, or enter the volume password.

[Delete all files?] Default: No

Enter "yes" if you wish to delete all the files in the directory.

The directory cannot be removed until all the files are deleted.

[Confirm while deleting?] Default: No

If you enter "yes", then you are prompted to confirm the deletion of each file in the directory by pressing GO, deny it by pressing CANCEL, or stop deleting files by pressing FINISH.

If you enter "no" or leave the field blank, then you are not prompted for confirmation and all the files are deleted.

RENAME

Introduction

The Rename command changes the name of a file.

A file can be renamed from directory to directory, or from volume to volume.

To rename more than one file, you must use the wild card feature. Wild cards must be used in the same way for the old and new name specifications.

Capitalization in file names can also be changed with this command. For example, myfile can be changed to MyFile.

This command is implemented internally by the Executive using the value !4.

Command Form

```
Rename
  File from          ______________________________
  File to            ______________________________
  [Overwrite ok?]    ______________________________
  [Confirm each?]    ______________________________
```

Parameter Fields

File from

Enter the file specification for the file or group of files you wish to rename.

You can use the wild card characters, the asterisk(*) or the question mark (?) in this field, but you must use them again in the same order in the File to field. The wild card characters can be used in the directory name.

Example:

```
Rename
  File from          a*b______________
  File to            c*d______________
  [Overwrite ok?]    _________________
  [Confirm each?]    _________________
```

File to

Enter the file specification for new file name(s).

If you wish to rename a file across a volume or directory remember to include a full file specification in this field.

If you used wild card characters in the file specification in the File to field, remember to use them again in the same order.

[Overwrite ok?] Default: Ask for Confirmation

If you enter "yes" and the file named in the File to field already exists, then the existing file is deleted and the rename operation continues.

If you enter "no" and the file named in the File to field already exists, the rename will be cancelled.

If you leave the field blank, the Executive will prompt you for confirmation before the existing file is deleted. You can confirm this by pressing GO, deny it by pressing CANCEL, or stop renaming files by pressing FINISH.

[Confirm each?] Default: No

If you enter "yes", then you are prompted to confirm each rename by pressing GO, deny it by pressing CANCEL, or stop renaming files by pressing FINISH.

If you enter "no" or leave the field blank, then you are not prompted for confirmation.

RESTORE

Introduction

The Restore command restores files onto a volume from an archive file created by either the Backup Volume or Selective Backup commands.

Restore can

- Restore an entire volume, creating any needed directories on the destination volume.

- Restore selected files to the same or different file specifications.

- Restore a file's characteristics (creation date, protection level, etc.) as they existed at the time of backup

- Optionally write the log of the Restore operation to a file or printer. (The log always appears on the screen.)

The volume onto which the files are to be restored can have been reinitialized with the IVolume command, or it can already have files on it. If the volume already contains a file with the same file specification as the one on the archive file, Restore optionally deletes and replaces it.

This command is implemented using the Run file Restore.Run.

Command Form

```
Restore
  [Archive file]                  ____________________
  [File list from]                ____________________
  [File list to]                  ____________________
  [Overwrite ok?]                 ____________________
  [Confirm each?]                 ____________________
  [Sequence number]               ____________________
  [Merge with existing file?]     ____________________
  [List files only?]              ____________________
  [Log file]                      ____________________
```

Parameter Fields

[Archive file] Default: [Archive]<sys>

Enter the name specified for the Backup Volume or Selective Backup commands.

An archive file can extend across more than one volume. When it does, it is split into files named [Archive]<Sys>.Ø1, [Archive]<Sys>.Ø2, etc.

[File list from] Default: All files

Enter a list of the files you wish to restore. The file specifications are in the following form:

<dirname>filename

No volume name is permitted.

If this field is defaulted, all the files on the archive file are restored.

The wild card character, the asterisk (*), can be used for dirname and filename.

To restore an entire directory, enter

<dirname>*

[File list to] Default: Original directory on default Volume

Enter a list of the file specifications that are to receive the restored files in the following form:

[volname]<dirname>filename

The volume name and directory name are optional. If only a file name is specified, the files are restored to the currently logged-in volume and directory.

If this field is defaulted, the files are restored to the same directory and file as when archived, but to the logged-in volume.

The wild card character, the asterisk (*), can be used for dirname and filename, and follows the same conventions as in the Executive.

[Overwrite ok?] Default: Ask for confirmation

If you enter "yes", an existing file of the same name is automatically deleted before a new one is restored.

If you leave this field blank, you are prompted for confirmation before an existing file is deleted.

If you enter "no" , the existing file is not overwritten.

[Confirm each?] Default: No

If you enter "yes", Restore prompts you for confirmation before the file is restored.

If you enter "no" or leave this field blank, Restore proceeds without user interaction.

[Sequence number] Default: 1

Enter the first volume with which you want to begin the Restore operation.

The default is 1, the first volume of the archive file.

If the Restore operation does not begin with the first volume, Restore creates any required directories 1Ø sectors in size and unprotected.

[Merge with existing file?] Default: No

If you enter "yes", the sectors of the target file are bypassed if the corresponding sector of the archive file is unreadable.

If you enter "no" or leave the field blank, the sectors of the target file are written with Ø's if any input/output error is detected on the archive file.

[List files only?] Default: No

If you enter "yes", the files on the archive file are listed; however, they are not restored.

[Log file] Default: Log appears on video

Enter the name of the file to which you want a report of the Restore operation written.

If the log file exists, the log is appended to it. If it does not exist, it is created.

If no log file is named, the log appears only on the screen.

Description

Operation. Restore first prompts you to mount the initial volume of the archive file. If no name is specified, Restore uses the default name [Archive]<Sys> and, if no sequence number is specified, it prompts you to mount the first archive file. For example, if no specifications are made, Restore displays this message:

```
Please mount [Archive]<Sys>.Ø1 (Press GO to
confirm, CANCEL to deny, or FINISH to return
to the Executive.)
```

Mount the appropriate volume and press GO. If Restore cannot find the volume or file (for example, a volume is mounted out of sequence), it prompts again.

Restore operates in either volume or selective mode.

Volume Restore. In volume mode, an entire archive file is restored to a volume. If the Restore form is defaulted, Restore copies all the files to the currently logged-in volume. Each file is restored with the same characteristics (creation date, protection level, etc.) it had when backed up.

Selective Restore. In selective mode, individual files or directories are restored. Specify the list of files in [File list from] in the Restore form. Specifying a list of files in [File list to] restores the files to a new destination. Defaulting [File list to] restores the files to the same directory and file names as when archived, but to the logged-in volume. Files specified in [File list from] must match in number the files in [File list to].

For example, the files from directory Work1 can be restored to a new directory, Work2. Specify <Work1>* in [File list from] and <Work2>* in [File list to].

Directories. The archive file created with the Backup Volume command contains directory information, while one created with the Selective Backup command does not.

When Restore encounters the directory information written by the Backup Volume command, it creates any needed directory with the same size, password, and protection level that it had when archived. If files are restored to a new directory, Restore makes it the same size, password, and protection level as the old directory.

If the restoration begins in the middle of the archive file set (with a sequence number other than Ø1), or the restoration is from an archive file created by the Selective Backup command, Restore has no directory information available to it. In this case, it creates any needed directory with a default size of 1Ø sectors and no protection level.

Note that in order for Restore to be able to create new directories, it must have the correct volume password (if one is specified).

Error Recovery. Restore attempts to continue past any input/output errors in the archive file.

If Restore encounters an error in the data portion of a file, or a record from the Backup Volume or Selective Backup commands that indicates the data was unreadable when archived, it records an indication of the failure in the log file, and the portion of the file that is in error is written as Ø's to the new file (unless you enter "yes" in the [Merge with existing file?] field).

If Restore encounters an error in the header portion of a file (which contains information about the directory name, the file name, and the size of the file), Restore creates, on the logged-in directory, a file named &Restore.n,

where n is a number on the logged-in directory, and displays this message:

> Unrecoverable file header information, creating &Restore.

This file contains all the data that can be recovered.

Backup Volume copies the list of directories twice, and Restore is usually able to create all the directories correctly from one of the copies. Even if both copies are unreadable, Restore creates default directories, if and when needed.

Specifying Sequence Number. Usually, each volume of the archive file is read in sequence starting with the first. However, when restoring selected files, it is possible to begin with the sequence number of the first volume of the archive file that contains one of the files you wish to restore. Though the restoration can start out of sequence, you must load each volume in sequence after that (for example: 3, 4, 5 or 7, 8, 9).

The directory information is maintained only on the first volume of the archive file. Thus, when the operation begins from a volume other than the first, Restore creates any required directory 1Ø sectors in size and unprotected. If 1Ø sectors are insufficient, create the necessary directory before beginning the Restore operation.

Restore Command Statistics Reporting. Restore reports the following statistics at the end of its operation:

- o Number of files processed: indicates the number of files that were read by Restore.

- Number of files successfully backed up: indicates the number of files that were successfully retrieved from the archive file and placed on the destination volume without errors.

- Number of files with input/output errors in data: indicates the number of files that were restored, but contained input/output errors in the data portion of the file.

- Number of files with missing header information: indicates the number of files that had unrecoverable header information. Unless you entered "yes" in the [List files only?] field, all such files are restored to temporary files &Restore.n.

- Number of files not successfully restored: indicates the number of files that could not be correctly restored to the destination volume. The causes include: invalid file specification, nonexistent volume, etc.

- Number of files not recoverable: indicates the number of files that could not be recovered. Input/output errors on the archive file rendered the entire header and data portion of the file unreadable.

Note that some files can belong in multiple categories. For example, a file with missing file header information can also have input/output errors in the data portion.

Examples

Example 1: Volume Restore

The form below, with all the fields left blank, restores all the files from the default archive file to the default volume (as set by the SignOn form or the Path command) :

```
Restore
[Archive file]
[File list from]
[File list to]
[Overwrite ok?]
[Confirm each?]
[Sequence number]
[Merge with existing file?]
[List files only?]
[Log file]
```

Restore first prompts you to mount volume 1:

```
Please mount [Archive]<Sys>.Ø1 (Press GO to
confirm, CANCEL to deny, or FINISH to return
to the Executive.)
```

If the wrong volume is mounted, it repeats the prompt. Restore creates all required directories on the default volume if they do not already exist. It then restores all files from [Archive]<Sys> to the default volume. As each volume of [Archive]<Sys> is restored, Restore prompts you to mount the next volume:

```
Please mount [Archive]<Sys>.Ø2 (Press GO to
confirm, CANCEL to deny, or FINISH to return
to the Executive.)
```

Example 2: Selective Restore to Old Specifications

The form below restores all files from the Work1 and Work2 directories of the archive file [Ø6Ø181]<Sys> to the same specifications they had when backed up:

```
Restore
  [Archive file]                  [Ø6Ø181]<Sys>
  [File list from]                <Work1>* <Work2>*
  [File list to]
  [Overwrite ok?]
  [Confirm each?]
  [Sequence number]
  [Merge with existing file?]
  [List files only?]
  [Log file]
```

Restore first prompts you to mount volume 1:

```
Please mount [Ø6Ø181]<Sys>.Ø1 (Press GO to
confirm, CANCEL to deny, or FINISH to return to
the Executive.)
```

If the wrong volume is mounted, it repeats the prompt. Restore then creates the directories Work1 and Work2 on the default volume if they do not already exist. It then restores the files from the Work1 and Work2 directories of [Ø6Ø181]<Sys> to the Work1 and Work2 directories of the default volume. As each volume of [Ø6Ø181]<Sys> is restored, Restore prompts you to mount the next volume:

> Please mount [Ø6Ø181]<Sys>.Ø2 (Press GO to confirm, CANCEL to deny, or FINISH to return to the Executive.)

Example 3: Selective Restore to New Specifications

The form below restores all the files from the Work1 directory to the X directory of the Win2 volume and all files from the Work2 directory to the Util directory of the default volume:

Restore

[Archive file]	
[File list from]	<Work1>* <Work2>*
[File list to]	[Win2]<X>* <Util>*
[Overwrite ok?]	
[Confirm each?]	Yes
[Sequence number]	3
[Merge with existing file?]	
[List files only?]	
[Log file]	[Spl]

Restore starts from the third volume of the default archive file, confirms each file before restoring it, and sends the log to the printer spooler.

Restore first prompts you to mount volume 3:

> Please mount [Archive]<Sys>.Ø3 (Press GO to confirm, CANCEL to deny, or FINISH to return to the Executive.)

If the wrong volume is mounted, it repeats the prompt. Restore then creates the directories X (on the Win2 volume) and Util (on the default volume), 1Ø sectors in size and unprotected, if they do not already exist. It then restores the files from the Work1 directory to the [Win2]<X> directory and the files from the Work2 directory to the Util directory on the default volume. Before restoring each file, Restore prompts:

```
Restore x?  (Press GO to confirm, CANCEL to
deny, or FINISH to return to the Executive.)
```

where x is the file specification of the file you wish to restore.

As each volume of [Archive]<Sys> is restored, Restore prompts you to mount the next volume:

```
Please mount [Archive]<Sys>.Ø4 (Press GO to
confirm, CANCEL to deny, or FINISH to return
to the Executive.)
```

Example 4: Restoring from Two Archive Disks

The form below restores all the files from two sets of archive files (with the default name Archive) to the default volume. Use the first archive file and leave all the fields blank (as in Example 1 above).

If the first archive file is restored without error, further restoration is unnecessary. However, if Restore encounters errors, use the second archive file and execute Restore again. Assume that Restore reported errors in the files M1 and M2 in the Work directory. Also assume that these files begin on the third volume of the archive file.

Mount the second archive file and complete the form as follows:

```
Restore
  [Archive file]                ____________________
  [File list from]              <Work>M1 <Work>M2
  [File list to]                ____________________
  [Overwrite ok?]               Yes
  [Confirm each?]               ____________________
  [Sequence number]             3
  [Merge with existing file?]   Yes
  [List files only?]            ____________________
  [Log file]                    ____________________
```

RESUME CLUSTER

Introduction

The Resume Cluster command reenables normal cluster operations. However, files that were open on the cluster workstations before cluster communications were disabled are not automatically reopened.

An application system must reopen the files for operation to resume without user intervention.

This command is implemented using the Run file ResumeCluster.Run.

Command Form

Resume Cluster

Parameter Fields

(This command has no fields.)

RUN

Introduction

The Run command invokes a user program. This command is used primarily to invoke programs not registered by means of the New Command command.

Run allows you to start any Run file, even if the correct Executive command form has not been built yet. When you use the Run command a variable length parameter block (VLPB)is built, which looks to the Run file as though the correct Executive form has been supplied.

The Run command is also used to run a program with the Debugger. If you press CODE-GO (instead of GO) after filling in the form, then the program is loaded by the operating system, but not started. Control is then given to the Debugger.

This command is implemented internally by the Executive using the value !18.

Command Form

Run
  Run file ________________________
  [Case] ________________________
  [Command] ________________________
  [Parameter 1] ________________________
  [Parameter 2] ________________________
  [Parameter 3] ________________________
        .
        .
        .
  [Parameter 16] ________________________

Parameter Fields

Run file

Enter the name of the Run file. A Run file consists of executable code that has been assembled and linked.

[Case] Default: 'ØØ'

Enter a two character case value. The Run file uses the case value to identify the command that invoked it. For more information on case see the parameter Management section of the CTOS Operating System Manual.

[Command] Default: No name

Enter the command name you wish to use to execute this Run file.

[Parameter 1] Default: None

Enter the parameter or subparameters for the first field you would include on the Executive command form for this command if one existed.

Continue to enter each parameter in order. You can enter up to sixteen parameters with this command.

Example

If you fill out the Run command form as shown below, the Executive will execute the command as if you had given the Format comand and specified the files abc and def to the file list prompt.

```
Run
  Run file         Format.run
  [Case]           ______________________
  [Command]        ______________________
  [Parameter 1]    abc def
  [Parameter 2]    ______________________
  [Parameter 3]    ______________________
        .
        .
        .
  [Parameter 16]   ______________________
```

SCREEN SETUP

Introduction

The Screen Setup command changes one or more of the screen attributes. Screen attributes set with the Screen Setup command remain in effect until you log out.

This command is implemented internally by the Executive using the value !17.

Command Form

```
Screen Setup
  [Reverse video?]
  [80 characters?]                                 __________
  [Screen length (default = full screen)]          __________
  [Suppress character attributes?]                 __________
  [Suppress pause between pages?]                  __________
  [Color, e.g. Green]                              __________
  [Screen time out (in minutes)]                   __________
                                                   __________
```

Parameter Fields

[Reverse video?] Default: No

> If you enter "yes", then dark characters on a light background are displayed.
>
> If you enter "no" or leave the field blank, light characters are displayed on a dark background.

[80 characters?] Default: No

> This field is only applicable if you have a model with wide screen capabilities.
>
> If you enter "yes", then the screen display becomes 80 columns wide.
>
> If you enter "no" or leave the field blank, the screen display is 132 columns wide.

[Screen length (default = full screen)]

Enter screen length desired in number of lines. Number of lines required for a full screen display varies with the model you are using.

[Suppress character attributes?] Default: No

If you enter "yes", then the field you type into is not reverse video when you fill in a command form.

[Suppress pause between pages?] Default: No

If you enter "yes", then the pause-between-pages facility is disabled.

If you enter "no" or leave this field blank the Executive prompts you to press NEXT PAGE or SCROLL UP before scrolling information off the screen that you may not have had time to read.

[Color, e.g. Green] Default: Green

This field is only applicable to color workstations.

Enter Green, White, Blue, Yellow, or Amber.

[Screen time out (in minutes)] Default: None

The screen time out feature turns the screen off when no characters have been typed for a specified number of minutes. You can turn the screen back on by pressing any key. The SHIFT key is recommended.

Enter the number of minutes to be used as the delay before the screen time out feature is activated.

SELECTIVE BACKUP

Introduction

The Selective Backup command copies individual files or directories from one volume to an archive file. (The Backup Volume command, in contrast, copies an entire volume.)

Selective Backup allows users to archive their personal files and requires only read access to the files being archived. Backup Volume, in contrast, requires device-level access.

Selective Backup can

o Archive individual files or directories.

o Archive only those selected files modified on or after a specified date, or date and time (incremental backup).

o Optionally write the log of the Selective Backup operation to a file or printer. (The log always appears on the screen.)

Selective Backup, unlike the Copy command, copies files to an archive file. In addition, Selective Backup operates more quickly than the Copy command.

This command is implemented using the Run file SelectiveBackup.Run.

Command Form

```
Selective Backup
  File list                                    ____________
  [Incremental from
    (e.g., Mon Jun 1 1981 8:ØØ pm)]            ____________
  [Confirm each?]                              ____________
  [Archive file]                               ____________
  [Delete existing archive file?]              ____________
  [Log file]                                   ____________
```

Parameter Fields

File list   Enter a list of the files you wish to back up. The list can include single files, directories, or sets of files (using wild cards).

Any file specification can include a password by appending

^<password>

to that file specification. For example:

<Work1>*^ABC

uses ABC as the password to access all the files in the Work1 directory. ABC is echoed as ###.

[Incremental from (e.g., Mon Jun 1 1981 8:ØØ pm)]

Enter the date from which files are to be backed up. Only files modified on or after the specified date are backed up.

A time can also be optionally specified after the year.

If no date is specified, all files are backed up. If no time is specified, all files from 12:ØØ am (midnight) of the specified day are backed up.

[Confirm each?] Default: No

If you enter " yes", Selective Backup prompts for confirmation before each file is backed up.

If you enter "no" or leave the field blank, the files are backed up without individual file confirmation.

[Archive file] Default: [Archive]<Sys>.nn

Enter the name of the archive file you wish to create.

If no file is specified, Selective Backup creates one named [Archive]<Sys>.nn.

Since the files being backed up usually occupy more space than is available on one floppy disk, [Archive file] can extend across more than one volume, split into files named [Archive]<Sys>.Ø1, [Archive]<Sys>.Ø2, etc. (with file extensions supplied by Selective Backup).

If a volume name other than Archive is specified, a directory name must also be specified so that the addition of a sequence number produces a valid file specification. For example, specify [Xyz]<Sys> to get [Xyz]<Sys>.Ø1, [Xyz]<Sys>.Ø2, etc., or [Xyz] <Sys>Abc to get [Xyz]<Sys>Abc.Ø1, [Xyz]<Sys>Abc.Ø2, etc.

[Delete existing archive file?] Default: No

If you enter "yes", any existing archive file is automatically overwritten.

If you enter "no" or leave this field blank, Selective Backup prompts if the archive file already exists:

```
File already exists. Delete?
(Press GO to confirm, CANCEL to
deny, or FINISH to return to the
Executive.)
```

[Log file] Default: Log is written to screen

Enter the name of the file to which you wish to write a report of the backup.

If the log file exists, the log is appended to it. If it does not exist, it is created.

If no log file is named, the log appears only on the screen.

Description

Operation. Before beginning a backup, use the IVolume command to initialize a supply of floppy disks. All these floppy disks must have the same volume name and no password. You can use any volume name that does not conflict with any currently in use. Or you can use the default volume name, Archive. The approximate number of disks required can be determined by giving a Files command for the list of files to be archived and entering "yes" in the [Details?] field. The Files command displays the number of sectors used. Divide this number by 800 to arrive at the approximate number of floppy disks required.

Selective Backup verifies the consistency of the specified parameters and opens the log file (if one is specified). It prompts you to mount the first volume of the archive file.

Mount the appropriate volume and press GO. Selective Backup creates the archive file. If the file already exists, Selective Backup overwrites it if you entered "yes" in the [Delete existing archive file?] field. Otherwise, it requests verification before overwriting the file.

Each file that you want to archive is opened and copied to the archive file. Selective Backup lists each file as it is archived. If any sectors of a file are unreadable, Selective Backup writes status information to the archive file. When the Restore command encounters this information, it reports that it cannot restore the specific sectors and continues operation.

Selective Backup does not archive system files such as FileHeaders.Sys, Mfd.Sys, etc.

Selective Backup, unlike the Backup Volume command, does not copy directory information to the archive file. When the archive file is restored, any needed directories are created with a default size of 10 sectors. If larger directories are needed, they must be created before the Restore operation begins.

Incremental Backup. Incremental backup saves time in the backup process. An incremental backup archives only the files created or modified on or after a specified date and time. The entire file (not just the modified portion) is copied to the archive file. Incremental backup is described in further detail under the Backup Volume command, above.

Examples

Example 1: Archiving an Entire Disk

The example below shows how to archive all the files on volume ABC to an archive file. This is similar to Backup Volume with no verification.

```
Selective Backup
  File list                             [ABC]<*>*___________
  Incremental from (e.g.,
    Mon Jun 1, 1981 8:ØØ pm)]           ____________________
  [Confirm each?]                       ____________________
  [Archive file]                        ____________________
  [Delete existing
    archive file?]                      ____________________
  [Log file]                            ____________________
```

Example 2: Archiving User Directories

The example below shows how to archive the user directory <Work2>.

```
Selective Backup
  File list                             <Work2>*____________
  Incremental from (e.g.,
    Mon Jun 1, 1981 8:ØØ pm)]           ____________________
  [Confirm each?]                       ____________________
  [Archive file]                        ____________________
  [Delete existing
    archive file?]                      ____________________
  [Log file]                            ____________________
```

Example 3: Archiving a List of Files

The example below shows how to archive a list of files to an archive file. The list is contained in the file "Myfile."

```
Selective Backup
  File list                             @Myfile_____________
  Incremental from (e.g.,
    Mon Jun 1, 1981 8:ØØ pm)]           ____________________
  [Confirm each?]                       ____________________
  [Archive file]                        ____________________
  [Delete existing
    archive file?]                      ____________________
  [Log file]                            ____________________
```

SET FILE PREFIX

Introduction

The Set File Prefix command sets the default file prefix added to file specifications when the volume and directory names are omitted.

The Executive Status frame displays the current default volume, directory and file prefix. If no file prefix is displayed, none has been set.

The file prefix remains in effect until you logout or remove the prefix by executing the Set File Prefix command again leaving the parameter field blank.

This command is implemented internally by the Executive using the value !9.

Command Form

Set File Prefix
  Default file prefix ______________________________

Parameter Fields

Default file prefix

Enter the character string you want to have prefixed to the file name.

Example

If you are logged-in to volume Accounting and directory PettyCash, and you execute the command form:

```
Set File Prefix
  Default file prefix Jones>
```

then the file specification Ø2-Ø2-8Ø is equivalent to the full file specification [Accounting]<PettyCash>Jones>Ø2-Ø2-8Ø.

SET PROTECTION

Introduction

The Set Protection command assigns a new protection level and, optionally, a file password to each file specified in the File List field.

For an explanation of the password protection system see the subsection "File Protection in Section 4, "Basic Concepts."

This command is implemented internally by the Executive using the value !1Ø.

Command Form

Set Protection
File List ____________
New protection level (e.g., 15) ____________
[New password] ____________
[Confirm each?] ____________

Parameter Fields

File list Enter a file specification , file list or at-file name.

New protection level (e.g., 15)

Enter the protection level you wish to assign. See Table 4-1 in Section 4, "Basic Concepts," for a list of the protection levels available.

[New password] Default: No password change

This field must be completed if the New protection level is one that requires a password. This field assigns file passwords only. Passwords can have a maximum of 12 characters.

[Confirm each?]   Default: No

If you enter "yes", then you are prompted for confirmation before the protection level of each file is set. You can confirm it by pressing GO, skip to the next file by pressing CANCEL, or stop setting file protection levels by pressing FINISH.

SET DIRECTORY PROTECTION

Introduction

The Set Directory Protection command allows you to add, change, or remove a directory password, and/or to change the default file protection level for files in that directory.

See the subsection "File Protection," in Section 4, "Basic Concepts" for more information on file protection levels and passwords.

This command is implemented using the Run file SetDirectoryProtection.Run.

Command Form

Set Directory Protection
Directory name (e.g., sys) ________________
[Volume or directory password] ________________
[New protection level (e.g., 15)] ________________
[New password] ________________

Parameter Fields

Directory name (e.g., sys)

Enter directory name.

[Volume or directory password]
Default: No password

If a directory password was assigned previously, you must enter it or the volume password to make a change to either the directory password or to the default file protection level for files in this directory.

If no directory password was assigned, you can leave this field blank.

[New protection level (e.g., 15)]
Default: No Change

If you want to change the default file protection level for this directory, enter the decimal value for the new protection level here. See Table 4-1 in Section 4, "Basic Concepts" for a list of the protection levels available.

[New password] Default: No Change

Enter the new directory password. Passwords can have up to 12 characters.

To remove a directory password, enter a pair of single quotes ('') in this field.

SET TIME

Introduction

The Set Time command sets the system clock. In a cluster workstation, this sets the system clock for the entire cluster configuration.

If the system clock has already been set, it can be modified by entering part of the date/time. The parts of the date/time field (described below) can be entered in any order, with or without spaces or punctuation.

This command is implemented using the Run file SetTime.Run.

Command Form

Set Time
  Date/Time (e.g., Mon Jun 1, 1981 8:ØØ pm) ______

Parameter Fields

Date/Time Enter the day of the week, month, date, year, and time. Options for each of these categories are explained below.

Day of Week

Enter the day of the week (e.g., Monday). Only enough of the name is required to make it unique from the other days of the week, the names of months, and the optional time qualifiers AM, PM, Noon, and Midnight. For example, "m" could be Monday, May, or Midnight, but "mo" uniquely specifies Monday.

Month Enter the current month (e.g., December). Only enough of the name is required to make it unique from the days of the week, the names of other months, and the optional time qualifiers AM, PM, Noon, and Midnight. For example, "m" could be Monday, May, or Midnight, but "ma" uniquely specifies May.

Day of Month

The day of the month is entered as a number between 1 and 31, 30, 29, or 28.

Year

The year can be entered with either two or four digits (the "19" is optional).

Abbreviated Month, Date, and Year

The month, date (day of month), and year can be combined into an abbreviated form (e.g., 6/1/82 for June 1, 1982).

Time

The time is the current hour and minute and an optional time qualifier: AM, PM, Noon, or Midnight.

If the optional qualifier is omitted, a time between 7:00 and 11:59 is interpreted as AM, while a time between 12:00 and 6:59 is interpreted as PM. To specify a time before 7:00 AM or after 7:00 PM, you must include AM or PM.

You can also use the 24-hour notation of two digits for the hour. For example, 09:21 is 9:21 AM, and 14:22 is 2:22 PM.

Examples

The following are examples for setting the system time to Monday, June 1, 1981, 8:00 AM:

- Mon June 1, 1981 8:00
- Mo Jun 1,81 8:00
- 6/1/81 Mo 8:00

The following are examples for changing the system time from 8:00 AM to 8:00 PM without changing the rest of the previously entered date and time:

- pm
- PM

The following are examples for setting the system date/time to Monday, December 5, 1982, 8:ØØ PM:

- 12/5/82mo8:ØØpm
- 5-Dec-82 2Ø:ØØ
- 8:ØØ Monday 5 December 1982 PM

The system has an internal calendar to ensure that entries are valid and consistent. For instance, since September 19, 198Ø, was a Friday, the system does not accept Thursday, September 19, 198Ø. It does not accept June 31 of any year and accepts February 29 only in leap years.

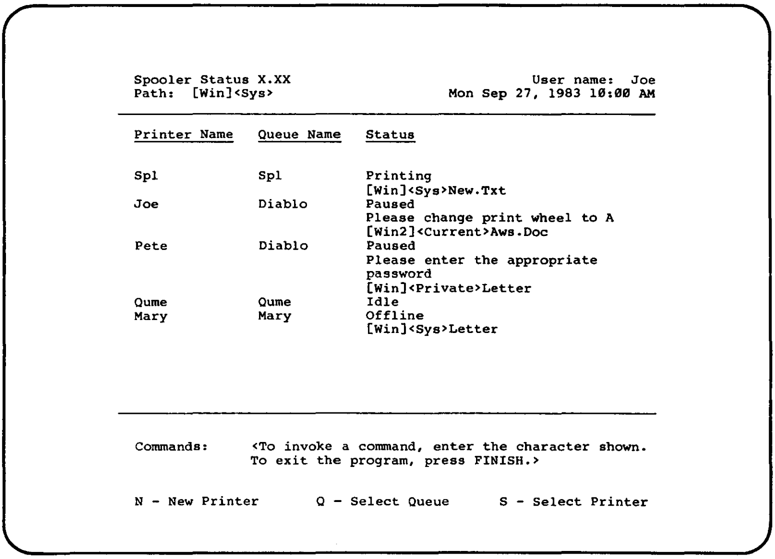
SPOOLER STATUS

Introduction

The Spooler command displays the status of printers and printer scheduling queues and provides the following subcommands:

A - Align Form
C - Cancel Print
D - Delete Print Request
E - Enter Password
F - Free Printer Channel
H - Halt Printer
M - Main Status Display
N - New Printer
P - Print File
Q - Select Queue
R - Restart Printer
S - Select Printer

When the Spooler command is first invoked, it displays a brief status of every printer in the system (either standalone or cluster configuration) as shown in Figure 6-14.



```
Spooler Status X.XX                                   User name:  Joe
Path:  [Win]<Sys>                               Mon Sep 27, 1983 10:00 AM
______________________________________________________________________

Printer Name    Queue Name    Status


Spl             Spl           Printing
                              [Win]<Sys>New.Txt
Joe             Diablo        Paused
                              Please change print wheel to A
                              [Win2]<Current>Aws.Doc
Pete            Diablo        Paused
                              Please enter the appropriate
                              password
                              [Win]<Private>Letter
Qume            Qume          Idle
Mary            Mary          Offline
                              [Win]<Sys>Letter



______________________________________________________________________

Commands:      <To invoke a command, enter the character shown.
               To exit the program, press FINISH.>

N - New Printer       Q - Select Queue       S - Select Printer
```

Figure 6-14. Status of All Printers Displayed after Giving the Spooler Command.

The user can then enter one of the following subcommands:

N New Printer, which places a specified printer under control of a printer spooler. Parameters include the name of the printer and its associated scheduling queue, printer configuration file, and priority of the printer spooler process.

Q Select Queue, which displays detailed status information for the specified scheduling queue and provides access to five subcommands.

This command is implemented using the Run file Spooler.Run.

A - Align Form

The Align Form subcommand is used after the printer pauses for any reason. You align the form and execute the Align Form subcommand. The printer reprints the first page and pauses again to allow you to align the form if necessary.

To restart the printer, you execute the Restart Printer subcommand. (See Restart Printer below.)

Subcommand Form

Align Form
<Press GO to execute, CANCEL to deny, or FINISH to exit.>

C - Cancel Print

The Cancel Print subcommand cancels the current printing request.

Subcommand Form

```
Cancel Print
  <Press GO to execute, CANCEL to deny, or FINISH
  to exit.>
```

D - Delete Print Request

The Delete Print Request subcommand deletes the queue entry for the specified print request from the scheduling queue.

Subcommand Form

```
Delete Print Request
File list  ________________________________________
  <Press GO to execute, CANCEL to deny, or FINISH
  to exit.>
```

Subcommand Parameter Fields

File list   Enter the list of file specifications previously specified in the print request.

E - Enter Password

The Enter Password subcommand allows a password to be entered when the printer pauses in response to the security mode request given in the Print command or the Print Files subcommand of the Spooler command. The file is not printed until you enter the password at the workstation to which the printer is connected.

Subcommand Form

```
Enter Password
  Password  ________________________________________
  <Press GO to execute, CANCEL to deny, or FINISH
  to exit.>
```

Subcommand Parameter Fields

Password   Enter a valid password for the file.

F - Free Printer Channel

The Free Printer Channel subcommand frees a printer channel from the printer spooler's control.

Subcommand Form

Free Printer Channel
  <Press GO to execute, CANCEL to deny, or FINISH to exit.>

H - Halt Printer

The Halt Printer subcommand halts the printer.

Subcommand Form

```
Halt Printer
  <Press GO to execute, CANCEL to deny, or FINISH
  to exit.>
```

M - Main Status Display

The Main Status Display subcommand redisplays the main status display shown in Figure 6-14 above.

Subcommand Parameter Fields

Main Status Display
<Press GO to execute, CANCEL to deny, or FINISH to exit.>

N - New Printer

The New Printer subcommand places a specified printer under control of a printer spooler and redisplays the main status.

Subcommand Form

New Printer
Printer channel ____________________
Printer name ____________________
Queue name ____________________
Printer configuration file ____________________
[Priority] ____________________
[Suppress banner?] ____________________
<Press GO to execute, CANCEL to deny, or FINISH to exit.>

Subcommand Parameter Fields

Printer channel

Enter the/ single-character printer channel code that specifies the printer channel to which the printer is connected:

Ø	is the parallel printer
A	is channel A
B	is channel B
C	is a channel on the CommIOP
.	.
.	.
.	.

Printer name

Enter the name of the new printer.

Queue name

Enter the name of the scheduling queue associated with the printer.

Printer configuration file

Enter the name of the Printer configuration file. This file describes the printer parameters and is created by the Create Configuration File command.

[Priority] Default: 10

Enter the priority at which you want the spooler process to run. Priority can be from 10-254, with 10 the highest. The default is 10. A priority lower than 128 (the default priority of the interactive application system) ensures that the printer spooler does not impact the interactive application system.

[Suppress banner?] Default: No

If you enter "yes", the banner page is not printed at the beginning of the file.

If you enter "no", the banner page is printed at the beginning of the file.

P - Print File

The Print Files subcommand causes a queue entry for printing of the specified list of files to be generated and added to the scheduling queue.

Subcommand Form

Print File
File list
[Number of copies]
[Delete after printing?]
[Special forms name]
[Print wheel name]
[Printing mode]
[Align form?]
[After date time]
[Security mode?]
[Priority]
[Confirm each?]
Press GO to execute, CANCEL to deny, or FINISH to exit.>

Subcommand Parameter Fields

File list — Enter a list of file specifications for the files you wish to print.

[Number of copies] — Default: 1

Enter the number of copies you want made of each file. The default is 1.

[Delete after printing?] — Default: No

If you enter "yes", the file is deleted.

If you enter "no", the file is not deleted.

[Special forms name] — Default: No special form

Enter the name (up to 12 characters) of the special paper on which you wish the file to be printed. If a name is specified, the printer pauses before printing so that the special paper can be loaded.

To restart the printer, use the Restart Printer subcommand.

[Print wheel name] Default: No special wheel

Enter the name (up to 12 characters) of the print wheel you want to use to print the file. If a name is specified, the printer pauses before printing so that the print wheel can be changed.

To restart the printer, use the Restart Printer subcommand.

[Printing mode] Default: Normal mode

Enter "N" for normal mode, "I" for image mode, or "B" for binary mode. (See the subsection "Printer Management" in Section 5, "Advanced Concepts," for more on print modes.)

[Align form?] Default: No

If you enter "yes", the printer pauses after the first page is printed so that the paper can be aligned. To restart the printer, use the Restart Printer subcommand.

If you enter "no" or leave the field blank, the printer does not pause for forms alignment.

[After date time] Default: ASAP

Enter the date and time after which you wish the file to be printed. The format is

```
Mon Jun 1, 1981 8:00 pm.
```

The default is to print as soon as possible.

[Security mode?] Default: No

If you enter "yes", the printer pauses before the file is printed. The file is not printed until you enter a password at the workstation to which

the printer is connected.
If you enter "no", the printer does not pause for a password before the file is printed.

[Priority] Default: 5

Enter the priority (Ø-9, with Ø the highest) at which you wish the file to be placed in the scheduling queue of the specified printer. The default is 5.

[Confirm each?] Default: No

If you enter "yes", you are prompted for confirmation before the file is formatted.

If you enter "no" or leave the field blank, formatting proceeds without user interaction.

Q - Select Queue

The Select Queue subcommand displays a detailed status of the files listed in the specified scheduling queue. (See Figure 6-15.)

To scroll to the end of the list when the list is longer than the screen, press NEXT PAGE.

To stop the listing, press CANCEL.

To redisplay the queue status, press PREV PAGE.

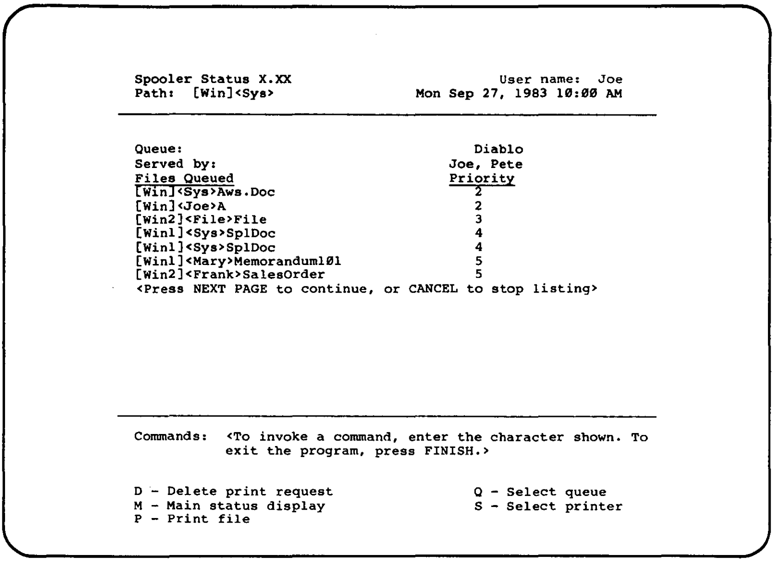
Select Queue provides five subcommands. When Delete Print Request or Print File is executed, the previous screen is restored.

Subcommand Form

Select Queue
Queue name ______________________________
<Press GO to execute, CANCEL to deny, or FINISH to exit.>

Subcommand Parameter Fields

Queue name — Enter the name of the scheduling queue for which you wish status information displayed.



```
Spooler Status X.XX                                  User name:  Joe
Path:  [Win]<Sys>                              Mon Sep 27, 1983 1Ø:ØØ AM

Queue:                                     Diablo
Served by:                              Joe, Pete
Files Queued                            Priority
[Win]<Sys>Aws.Doc                          2
[Win]<Joe>A                                2
[Win2]<File>File                           3
[Win1]<Sys>SplDoc                          4
[Win1]<Sys>SplDoc                          4
[Win1]<Mary>Memorandum1Ø1                  5
[Win2]<Frank>SalesOrder                    5
<Press NEXT PAGE to continue, or CANCEL to stop listing>

Commands:  <To invoke a command, enter the character shown. To
           exit the program, press FINISH.>

D - Delete print request                   Q - Select queue
M - Main status display                    S - Select printer
P - Print file
```

Figure 6-15. Status of the Scheduling Queue Displayed after Giving the Select Queue Subcommand.

R - Restart Printer

The Restart Printer subcommand restarts the printer after it has paused.

Subcommand Form

Restart Printer
  [Restart from page number] ______________________
  <Press GO to execute, CANCEL to Deny, or FINISH to exit.>

Subcommand Parameter Fields

[Restart from page number]   Default: From stop

Enter the page number from which you wish the printer to restart printing. The default is to restart printing where the printer stopped. To restart the printer from the beginning of the current page enter a "Ø".

S - Select Printer

The Select Printer subcommand displays a detailed status of the specified printer and of the files listed in the associated scheduling queue. (See Figure 6-16 below.)

To scroll to the end of the list when the list is longer than the screen, press NEXT PAGE. To stop the listing, press CANCEL.

To redisplay the printer status, press the PREVPAGE key.

Select Printer provides a list of subcommands that allows the user to enter printer and queue control instructions. After execution of a subcommand (except Free Printer Channel, Main Status Display, New Printer, Select Queue, and Select Printer), the previous screen is restored.

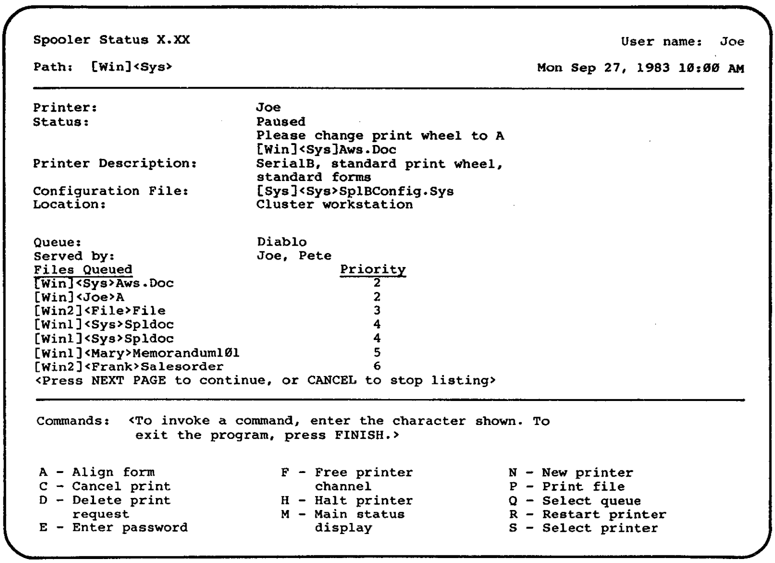
Subcommand Form

Select Printer
  Printer name ______________________________
  <Press GO to execute, CANCEL to deny, or FINISH to exit.>

Subcommand Parameter Fields

Printer name

Enter the name of the printer for which you want status information displayed.



```
Spooler Status X.XX                                               User name:  Joe

Path:  [Win]<Sys>                                      Mon Sep 27, 1983 10:00 AM
________________________________________________________________________________
Printer:                   Joe
Status:                    Paused
                           Please change print wheel to A
                           [Win]<Sys]Aws.Doc
Printer Description:       SerialB, standard print wheel,
                           standard forms
Configuration File:        [Sys]<Sys>SplBConfig.Sys
Location:                  Cluster workstation


Queue:                     Diablo
Served by:                 Joe, Pete
Files Queued                        Priority
[Win]<Sys>Aws.Doc                      2
[Win]<Joe>A                            2
[Win2]<File>File                       3
[Win1]<Sys>Spldoc                      4
[Win1]<Sys>Spldoc                      4
[Win1]<Mary>Memorandum101              5
[Win2]<Frank>Salesorder                6
<Press NEXT PAGE to continue, or CANCEL to stop listing>
________________________________________________________________________________

Commands:  <To invoke a command, enter the character shown. To
            exit the program, press FINISH.>

A - Align form              F - Free printer           N - New printer
C - Cancel print                channel                P - Print file
D - Delete print            H - Halt printer           Q - Select queue
    request                 M - Main status            R - Restart printer
E - Enter password              display                S - Select printer
```

Figure 6-16. Status of a Printer and Scheduling Queue Displayed after Giving the Select Printer Subcommand.

Introduction

The Stop Record command is given when you wish to stop recording keystrokes in a Submit file. See the subsection "Using a Submit File" in Section 5, "Advanced Concepts."

This command is implemented internally by the Executive using the value !20.

Command Form

Stop Record

Parameter Fields

This command has no fields.

SUBMIT

Advanced Command

Introduction

The Submit command reads characters directly from a Submit file, rather than from the keyboard. It is used to repeat a sequence of commands.

Submit can be used recursively, that is a Submit command can be included in a Submit file.

A Submit file can contain escape sequences, which invoke special functions.

For information on how to create a Submit file and use escape sequences, see the subsection "Using a Submit File" in Section 5, "Advanced Concepts."

For more information on the Submit facility, see the section on keyboard management in the <u>CTOS Operating System Manual</u>.

This command is implemented using the Run file Submit.Run.

Command Form

```
Submit
  File List              ______________________________
  [Parameters]           ______________________________
  [Force expansion?]     ______________________________
  [Show expansion?]      ______________________________
```

Parameter Fields

File list — Enter the name of the Submit file(s) from which characters are read. Files are read sequentially.

[Parameters] — Default: No parameters

Enter any parameters that replace escape sequences within the Submit file named in File list, above.

Escape sequences are indicated within the Submit file with the percent sign (%).

When a percent sign (%) is followed by

a number Ø-9, the submit facility takes that escape sequence as a variable, to be replaced by parameters supplied from the [Parameters] field.

The first parameter listed replaces all instances of %Ø, the second, %1, and so on through %9.

To replace an escape sequence with a null string you can enter the parameter as a pair of single quotes (''). To enter a string of characters containing spaces, you must enclose the string in single quotes.

[Force expansion?] Default:No

A file is automatically expanded when one of the items below applies:

- more than one file is named
- parameters are specified
- the first character in the named file is %
- you entered "yes" in the [Show expansion] field

Forcing expansion creates a temporary file containing the expanded characters with outstanding characters from any recursive calls to Submit appended to it.

The temporary file is then opened in read mode, the file handle is marked long-lived, and the file is given to the operating system as the source for character input of the SysIn facility (the System Input Manager). If the SysIn facility already had a file for character input (because of a recursive call), that file is closed.

The temporary file is located either in the $ directory of the System volume,[Sys]<$>, or in the logged-in volume and directory. Since recursive calls are permitted, two temporary files are tried in each directory. Submit tries the temporary files in the following order:

```
[Sys]<$>Submit0.Tmp
[Sys]<$>Submitl.Tmp
Submit0.Tmp
Submitl.Tmp
```

If a recursive call to Submit produces an infinite expansion, the temporary files expand to fill all available disk space, and Submit terminates with the status code "Disk full" or "Too many runs".

[Show expansion?] Default:No

If you enter "yes", the expansion of the file, showing variables replaced with the parameters you have supplied, appears on the screen.

This feature is especially useful for debugging complicated conditional expansions.

TYPE

Introduction

The Type command displays each of the specified files on the screen.

This command is implemented internally by the Executive using the value !6.

Command Form

Type
  File list ______________________
  [Confirm each?] ______________________

Parameter Fields

File list — Enter the file specification, file list, or at-file name for the file(s) you wish to see displayed.

[Confirm each?] — Default: No

If you enter "yes", then you are prompted for confirmation before each file is displayed. You can confirm it by pressing GO, skip to the next file in File list by pressing CANCEL, or stop displaying files by pressing FINISH.

If you enter "no" or leave this field blank, then you are not prompted for confirmation.

Introduction

The User File Editor command creates and modifies User Configuration files. With the User File editor you can easily add new users to the system, specify which applications each user sees initially after SignOn, and modify existing User Configuration files.

The User File Editor lets you select a user environment and other options to be included in the User Configuration file. Environment choices are offered through one menu and additional options through another.

The initial environment is the application loaded after SignOn. In some cases this may be your total environment. For example, you may go directly into the Word Processor after SignOn and return to SignOn immediately when you FINISH. In other cases, an additional application may be invoked when you exit from the first. You might, for instance, want the Executive invoked after you FINISH the Word Processor.

The User File Editor uses a Template file to define the options available to you for user configuration characteristics. Since the appearance of the User File Editor is highly dependent on the Template file, the menus may appear differently on your screen than those described here. See your system administrator if you need to clarify how to use your version of the User File Editor.

For more information on User Configuration files and the Template file see the subsection "User Configuration" in Section 5, "Advanced Operations."

The User File Editor Command form and parameter fields are described below. For a discussion of each menu see the section following the form.

This command is implemented using the Run file UserFileEditor.Run, a Template file, and an Executive Command file.

Command Form

User File Editor
User name ____________________
[Old password] ____________________
[New password] ____________________
[User template file] ____________________
[Command file] ____________________

Parameter Fields

[User name] Default: None

Enter the name you want to assign for this user. You can use any keyboard characters. The user name cannot be more than 30 characters long. You cannot use a wild card character in this field.

The user name is used when you sign on. User names must be unique. A person may have more than one user name, or a user name can be shared.

The name you enter is used to create a User Configuration file on the Sys volume and Sys directory, with the format [Sys]<Sys>Username.User.

[Old password] Default: Logged-in Password

Enter a valid password for the User Configuration file you want to access. You must give a password to access the file if it is password protected. The password entered may be a volume or file password.

If you leave this field blank, the password you signed on with or gave in the Path command is used. If that password is not a valid one for this user, an error message, "Error 219, Access denied", will appear on the screen.

[New password] Default: No new password assigned

Enter the password you want to assign to the User Configuration file. The password should be a volume or file password.

If you give a password for a User Configuration file, the protection level 51, Nondirectory Password, which requires a volume or a file password, is assigned. If the system release does not support protection level 51, protection level 19 is assigned.

[User template file]

Default: [Sys]<sys>userfiletemplate.sys

Enter the name of the Template file to be used by the User File Editor. The Template file contains the description of the system options from which you can choose.

[Command file]

Default: Current Exec Command File or [Sys]<sys>Sys.cmds

Enter the file specification for the Command file you want the User File Editor to access for this user.

Description

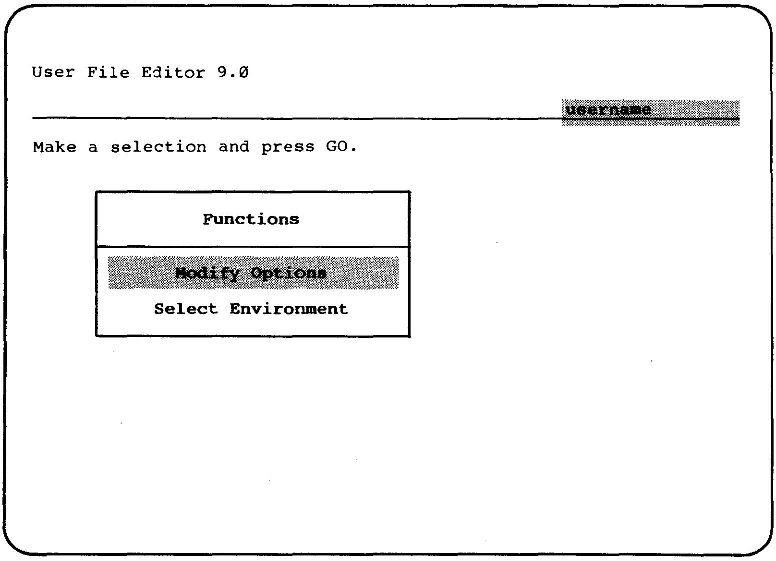
To use the User File Editor you can fill out the fields of the form and press GO. If you specified a new user name, the User File Editor asks you if it should create a new User Configuration file. If the user name already exists, the User File Editor accesses the existing User Configuration file for you.

When a new file is created, a set of defaults, which are defined in the Template file, are set up for you. Therefore, even though you may not edit any fields, a valid User Configuration file can still be created.

The user name for the configuration file you are editing is displayed and highlighted at the top right-hand corner of the screen.

The User File Editor uses a special function key, f2 or "Undo." The Undo key returns the previous parameter values to the field you are currently editing.

Main Menu. A Main Function menu is displayed from which can select the Environment menu or the Modify Option menu. Figure 6-17 shows the Main menu. You can pick a function by moving the highlight bar and pressing GO. If you want to choose the default environment or default options, then you do not need to display that menu at all.



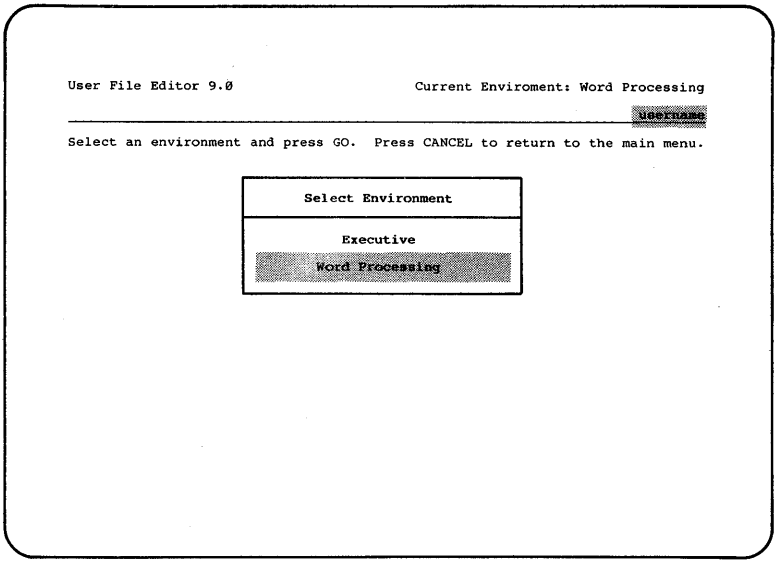
User File Editor 9.Ø

username

Make a selection and press GO.

Figure 6-17. User File Editor Main Menu.

Environment Menu. The Environment menu lists the choices for the initial application that the user sees after SignOn. An example of an Environment menu is shown in Figure 6-18. You can pick one environment by moving the highlight bar and pressing GO. Some application environments have no edit options. If so, a message tells you this and you press GO to confirm this change of environment.



User File Editor 9.0 Current Enviroment: Word Processing

username

Select an environment and press GO. Press CANCEL to return to the main menu.

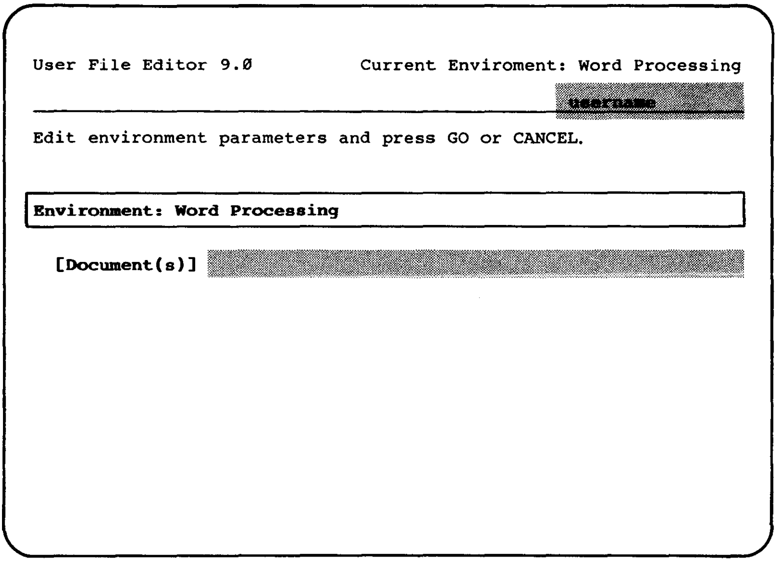
Figure 6-18. User File Editor Environment Menu.

If the environment you choose is one that has parameters, then the parameter prompts are displayed for you. If these fields are required by the application system you must fill them in. Optional parameter fields have square brackets ([]) around the parameter prompts.

The environment currently included in the User Configuration file is displayed in the upper right-hand corner of the screen. Figure 6-18 shows "Executive."

If you are editing a file not originally created with the User File Editor, the name displayed for the environment is "Unnamed." When you use the User File Editor to choose an environment for this file, the display changes to a name listed in the Environment menu.

If you select an environment that has parameters, an environment form will be displayed. Figure 6-19 shows a sample Environment entry form.



```
User File Editor 9.Ø          Current Enviroment: Word Processing
                                                       username
Edit environment parameters and press GO or CANCEL.

Environment: Word Processing

  [Document(s)]
```

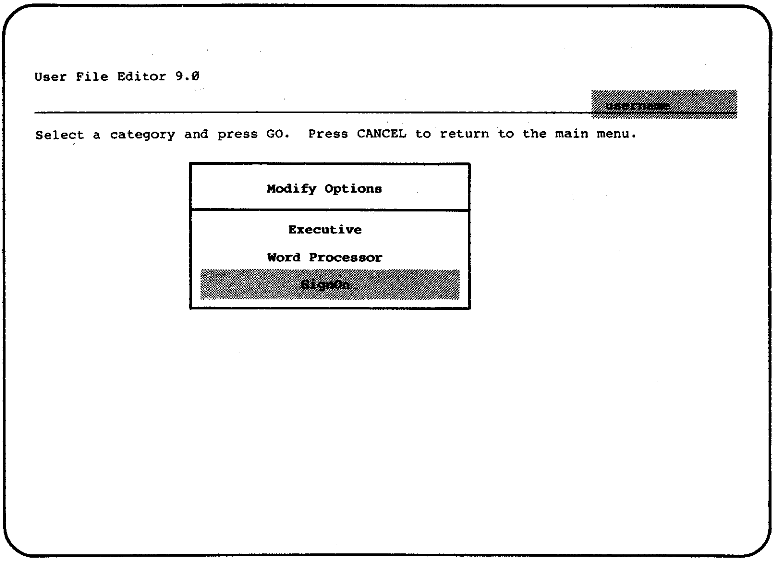
Figure 6-19. Environment Entry Form.

Modify Options Menu. If you request the Modify Options menu from the Main menu, the User File Editor displays a list of categories. Figure 6-2Ø shows the Modify Options menu. You can pick one by moving the highlight bar and pressing GO.

Once you select a category an option entry form appears. You edit the User Configuration file entries by filling in this form. Figure 6-21 shows an option entry form.

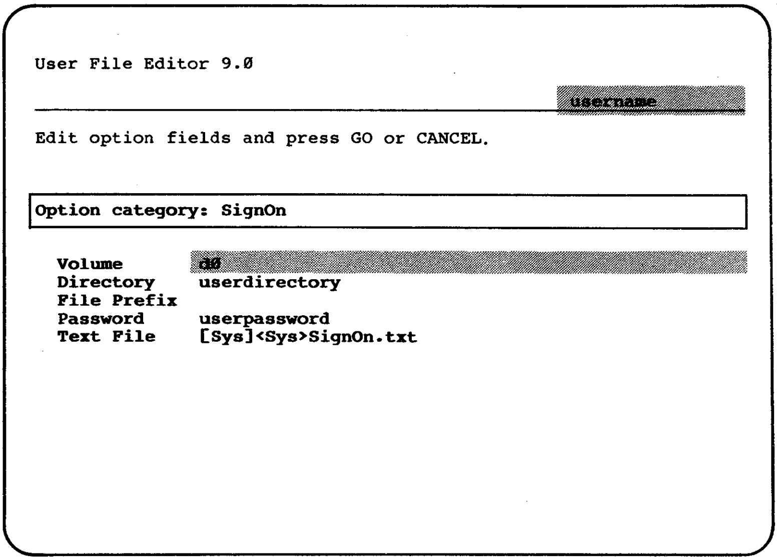
The option entry form includes the name of the category, the descriptions of the User Configuration file fields, and previously set or default values for those fields. If a field currently shows a value, you can change that value by typing the new one into that field. If fields are left blank, system applications use their own default values. When you have completed the form, you can press GO to return to the option menu.

You can choose another category and can return to any category as often as you like.



```
User File Editor 9.0
                                                          username
Select a category and press GO.  Press CANCEL to return to the main menu.
```

Figure 6-20. Modify Options Menu.



```
User File Editor 9.0
                                                   username
Edit option fields and press GO or CANCEL.

Option category: SignOn

  Volume        d0
  Directory     userdirectory
  File Prefix
  Password      userpassword
  Text File     [Sys]<Sys>SignOn.txt
```

Figure 6-21. User File Editor Option Entry Form.

To leave the User File Editor, press FINISH. You are prompted to press GO to save the file as edited. You can press CANCEL to discard all edits.

Error Messages. Most error messages that occur within the User File Editor are self-explanatory.

VOLUME STATUS

Introduction

The Volume Status command displays the status of the specified disk volume or of the volume mounted on the specified disk device. The displayed information includes

- the creation date
- the last modification date
- the number of free pages
- the number of free file headers
- a list of the directories

This command is implemented using the Run file VolumeStatus.Run.

Command Form

Volume status
[Volume or device name (e.g., Accounting)] ____

Parameter Fields

[Volume or device name (e.g., Accounting)]

Default: Logged-in Volume

Enter the name of the volume for which to display status information.

BATCH MANAGER

The Batch Manager allows users to execute batch jobs noninteractively under control of job control language (JCL) files. The Batch Manager executes in the same partition as the application it is controlling. The Batch Manager is replaced by an application Run file in each job step.

The Batch Manager can be executed either in the primary or in a secondary partition. It can be executed concurrently in any number of secondary partitions. (For more information on partitions, see the section on partition management in the *CTOS Operating System Manual*.)

BATCH JOBS AND JOB STEPS

A *batch job* is a JCL file that is processed under control of the Batch Manager. The JCL file directs the Batch Manager to load and activate a sequence of Run files to accomplish a single function. Parameters (data) can be passed from the Batch Manager to the Run files.

A *job step* is a single Run file executed in one step of the batch job.

For example, a batch job can produce an executable Run file from source code written in a high-level language. The job steps execute Run files that compile, link, and test the code.

Another example of a batch job is a weekly update of customer files. The job steps sort input records by account number, update customer files, and prepare management reports.

JOB CONTROL LANGUAGE FILES

A *JCL file* is a text file containing JCL statements that direct the Batch Manager in processing batch jobs. Each JCL file contains a single batch job.

JCL statements control operations such as loading and activating Run files, and passing run-time parameters to the Run files.

Figure 7-1 shows an example of a JCL file.

```
$JOB PascalTest,Pascal,,PascalTest.SysOut
;
;  This job compiles, links, and runs a Pascal program.
;
$RUN [Sys]<Sys>PascalFE.Run,PascalTest.Pas
$RUN [Sys]<Sys>Linker.Run,PascalTest.Obj,PascalTest.Run
$RUN PascalTest.Run
;
$END
```

Figure 7-1. Example of a JCL File.

```
$JOB PascalTest,pascal,,PascalTest.SysOut
Execution begun: Wed May 26, 1982 7:23 PM
;
;  This job compiles, links and runs a Pascal program.
;
$RUN [Sys]<Sys>PascalFE.Run,PascalTest.Pas
VNPascal 6.Ø1
 Pass One    No Errors Detected.
VN
 Code Area Size = #ØØ64 (   1ØØ)
 Cons Area Size = #ØØ4A (    74)
 Data Area Size = #ØØØ2 (     2)

 Pass Two    No Errors Detected.
Termination status code: Ø

$RUN [Sys]<Sys>Linker.Run,PascalTest.Obj,PascalTest.Run
Linker 6.Ø1
Termination status code: Ø

$RUN PascalTest.Run
VNPascal installation has been verified.
Termination status code: Ø

$END

Execution ended: Wed May 26, 1982 7:25 PM.
```

Figure 7-2. Example of JCL File Output.

When the JCL file shown in Figure 7-1 is executed, the information shown in Figure 7-2 is sent to the file PascalTest.SysOut.

BATCH PROCESSING IN A SECONDARY PARTITION

The Batch Manager, when executing in a secondary partition, processes a queue of JCL files that is managed by the Queue Manager. The Batch Manager constantly polls the Queue Manager to obtain the next JCL file to be processed. The Batch Manager processes the JCL file specified in the queue entry returned from the Queue Manager.

Since the application system executing in the primary (interactive) partition is using the keyboard, the job step executing in the secondary partition cannot use the keyboard for input. An application system using the keyboard when executing in the primary partition must be able to obtain input from some other source when executing in the secondary partition. The SysIn byte stream facility permits keyboard byte streams to read data embedded in the JCL file.

Similarly, an application system executing in a secondary application generally cannot send output to the screen since the application system executing in the primary partition uses the screen to interact with the user. The SysOut byte stream facility redirects an application system's video byte stream output to a disk file or to the printer spooler. The SysOut byte stream stays open during the processing of the entire batch JCL file. The SysOut byte stream is passed from job step to job step, each job step appending its output to the SysOut output file.

Each active Batch Manager maintains a <u>Batch Log file</u>. The log file is the Batch Manager's record of all the jobs it has processed in the order they were processed. In this log file, the Batch Manager records the time it was installed, the time each job started, the time each job finished, and the termination status code for the last job step executed. A job can also write to the log file with the $LOG JCL statement. (See the subsection on "Job Control Language Statements.")

You interface with a Batch Manager in a secondary partition through two Batch commands: Batch

Status and Batch. You submit and remove JCL files and cancel a job step or batch job by using subcommands of the batch status command. (See Section 6, "Commands.")

The Batch Status command allows a user at a cluster workstation to submit a job to a Batch Manager located at another workstation, cancel a job executing at any cluster workstation, or remove the Batch Manager's secondary partition for any Batch Manager in the cluster configuration. Figure 7-3 shows an example of a cluster configuration with noninteractive batch processing.

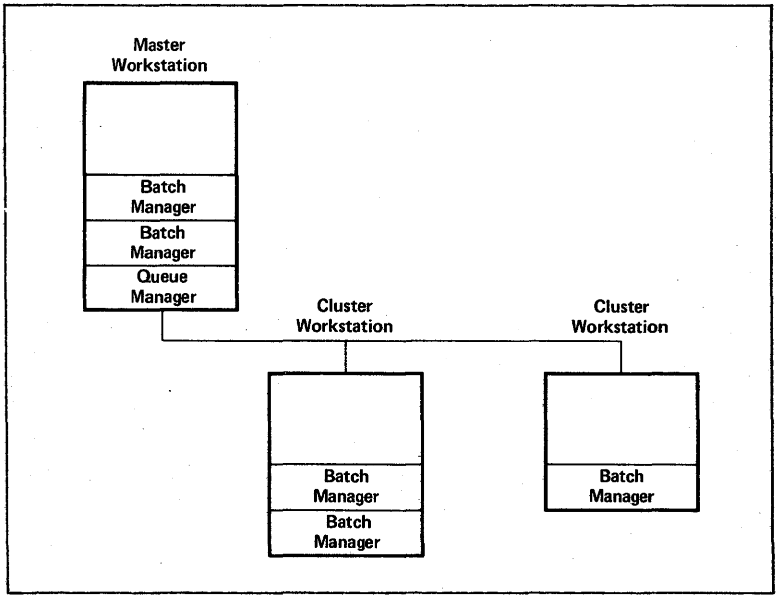


Figure 7-3. Example of a Cluster Configuration with Noninteractive Batch Processing.

Note that a batch job must be able to access all necessary files. Thus, all files needed by the batch job should be located at the master workstation. To use files located on a local file system, follow the constraints described in the "Batch Scheduling Queue" subsection. Also, note that the user profile file specified in the $JOB statement must reside at the master workstation. (See the subsection "Job Control Language Statements.")

Typical Batch Sequences

A typical sequence is given below for installing a Batch Manager in a secondary partition.

1. The system administrator edits the [Sys]<Sys>Queue.Index file to include the queues that are required for batch processing. These include the desired scheduling queues, a control queue for each Batch Manager to be installed, and one status queue.

2. The system administrator installs the Queue Manager in the master or standalone workstation.

3. The system administrator uses the Install Batch command to create a secondary partition and install a Batch Manager it.

A typical sequence is given below for submitting a batch job for processing by a Batch Manager in a secondary partition:

1. The user creates a JCL file using the Editor or Word Processor.

2. The user invokes the Batch Status command to submit the JCL file for batch processing.

3. The Batch Status command generates a queue entry describing the JCL file to be processed. It requests the Queue Manager to queue this entry in the appropriate scheduling queue.

A typical sequence is given below for the processing of a JCL file by a Batch Manager in a secondary partition:

1. The Batch Manager obtains a queue entry from the Queue Manager that describes the net JCL file to be processed.

2. The Batch Manager records start-up information for the JCL file in the batch log file.

3. The Batch Manager executes the job steps specified in the JCL file.

4. The Batch Manager records termination information for the JCL file in the batch log file.

5. The Batch Manager polls the Queue Manager to obtain a queue entry describing the next JCL file to be processed.

INSTALLING THE BATCH MANAGER

You invoke the Install Batch command to create a secondary partition and then install the Batch Manager in it. You can create as many batch partitions as memory limitations allow and can specify the size of the batch partition to be created. (See the Install Batch command in Section 6, "Commands.")

To automatically install a Batch Manager in a secondary partition when the system is bootstrapped, the following statement is placed in the SysInit JCL file.

```
   $RUN [Sys]<Sys>InstallBatch.Run,[size],[name] &
       ,[partition],[priority]
```

where

size — is the (optional) memory size of the batch partition to be created. The default is 2ØØK.

name — is the (optional) name of the batch scheduling queue from which the Batch Manager obtains JCL files to process. If name is not specified, the default scheduling queue is BATCH.

partition — is the (optional) name of an existing partition into which the Batch Manager is to be installed. When used in the SysInit.JCL file, partiion should not be specified.

The default is BatchPartØØ. (See the Install Batch command in Section 6, "Commands.")

priority — is the (optional) priority at which the Batch Manager executes. The default priority is 129.

BATCH QUEUES

When installing the Queue Manager and creating the queues required for the operating system, the system administrator creates three types of batch queues:

- a batch scheduling queue for each Batch Manager
- a batch control queue for each Batch Manager
- a batch status queue for the cluster configuration

The three types of batch queues are discussed below. For more information on queue management, see the section on queue management in the CTOS Operating System Manual.

BATCH SCHEDULING QUEUE

Each batch scheduling queue contains queue entries that describe batch JCL files to be processed. When you invoke the Batch Status command and submit a batch JCL job for processing by a Batch Manager, the Batch Status command generates a queue entry that describes the specified JCL file. The Batch Status command then sends this queue entry to the Queue Manager, which inserts it into the batch scheduling queue. The entry contains, for example, the JCL file name, job name, and current logged-in user name. (Table 7-1 shows the format of a batch scheduling queue entry.)

When a Batch Manager is installed in a secondary partition, it is passed as a parameter the name of the batch scheduling queue from which queue entries (describing JCL files to be processed) are obtained. Each Batch Manager constantly polls the Queue Manager to obtain the next JCL file to be processed.

A single batch scheduling queue can be served by several Batch Managers on different workstations. Since all the Batch Managers are constantly polling the Queue Manager to obtain the next available queue entry, you cannot determine which Batch Manager will actually process the JCL file. Therefore, all Batch Managers obtaining entries from this batch scheduling queue must be able to access all the files required to execute a job.

For example, if the scheduling queue BATCH is served by Batch Managers in the master workstation and cluster workstation A, you should not submit a job from cluster workstation B, which requires files located on the local file system at cluster workstation B. A Batch Manager located at the master or cluster workstation A cannot access files located at cluster workstation B. Therefore, unless the user can guarantee a submitted JCL file will be processed at its own local workstation, all files required should be located at the master workstation.

You can guarantee a job is processed on a particular workstation by making sure that the Batch Manager executing in the chosen workstation is the only one serving a particular batch queue. For example, Joe wants all his batch jobs to run on the cluster workstation located in his office so that they can access files on his local file system. Joe includes a batch scheduling queue named JOESBATCH in the [Sys><Sys>Queue.Index file. Joe then installs a Batch Manager at his cluster workstation with the Install Batch command, specifying JOESBATCH as the scheduling queue for the Batch Manager to operate on. As long as the Batch Manager in Joe's cluster workstation is the only one serving the batch scheduling queue JOESBATCH, all the jobs Joe submits to the JOESBATCH queue are processed at his workstation.

Table 7-1. Format of a Batch Scheduling Queue Entry.

Offset	Field	Size (bytes)	Description
Ø	SbJCLfilespec	91	File specification for a JCL file.
91	SbJobName	13	Job name for a scheduled JCL file. This must match job name contained in $JOB statement.
1Ø4	timeSubmitted	4	The time the job was submitted in date/-time format.
1Ø8	SbUserName	31	Name of user who submitted the job.
139	bManagerNum	1	Identification number of the Batch Manager that is processing the job. This should be equal of Offh when the job is submitted and will be rewritten by a Batch Manager.
14Ø	cParams	1	Number of parameters passed to the job.
141	rgsbParams		Packed array of sb strings that are the passed parameters starting at Ø

BATCH STATUS QUEUE

A batch status queue named BatchStatus contains detailed status information about all the active Batch Managers in the operating system. This information includes the job being processed by the Batch Manager, the job step currently executing, the name of the user who submitted the job, and so forth. (See Table 7-2 below.) You can display this information with the Batch Status command.

Each active Batch Manager marks a queue entry in the BatchStatus queue and constantly updates this entry to reflect the current state of the Batch Manager. An application program reads these entries with the ReadNextQueueEntry operation. (See the subsection on Queue Management in the CTOS Operating System Manual.) Those entries with the ServerUserNum field of the returned Queue Status Block not equal to OFFFFh belong to active Batch Managers.

BATCH CONTROL QUEUE

A separate batch control queue is defined for each Batch Manager by the system administrator. The system administrator must define enough batch control queues to accommodate the maximum number of Batch Managers to be installed in the cluster configuration at any one time. (See Table 7-3 below.)

The name of each batch control queue is Batch Control appended by two digits (ØØ to nn, where nn is the maximum number of Batch Managers to be active concurrently in the cluster configuration). Examples are BatchControlløØ, BatchControllØ1, and BatchControllØ2.

The two digits are also used as the Batch Manager identification number.

Each Batch Manager maintains its own batch control queue for internal control purposes.

BATCH MANAGER IDENTIFICATION NUMBER

When installed, the Batch Manager establishes itself as a server of the batch scheduling queue, the batch status queue, and the next free batch control queue (which establishes the Batch Manager identification number). The Batch Manager also marks an entry in the status queue that corresponds to the Batch Manager identification number.

The Batch Manager identification number is used by the Batch Status command.

Table 7-2. Format of a Batch Status Queue Entry. (Page 1 of 2)

Offset	Field	Size (bytes)	Description
Ø	sbBatchQueue	51	Batch queue being served by the Batch Manager.
51	SbJobName	13	Job currently being processed. If SbJobName is Ø, the Batch Manager is idle.
64	sbJobStep	133	Current job step being executed.
197	CbManagerIdNum	1	Always equal to 1. This is used with the ReadKeyedQueueEntry operation.

Table 7-2. Format of a Batch Status Queue Entry. (Page 2 of 2)

Offset	Field	Size (bytes)	Description
198	bManagerNum	1	Identification number of the Batch Manager that is processing the job. This should be equal to OFFh when the job is submitted and will be rewritten by a Batch Manager.
199	timeStarted	4	The time the current job was started in date/time format.
2Ø3	SbUserName	31	Name of the user that submitted the job being processed.
234	JobStepIdentifier	4	This must be specified in the control queue entry to cancel a job step.

Table 7-3. Format of a Batch Control Queue Entry.

Offset	Field	Size (bytes)	Description
Ø	bCommand	1	1 = cancel the current batch job. 2 = cancel the current job step. 3 = remove the batch partition. 4 = print the Batch Manager's log file.
1	sbJobName	13	Job name of the job currently being processed. This must be specified for the cancel batch job and cancel job step operations.
14	JobStepIdentifier	4	This is the job step identifier found in the Batch Manager's status queue entry. This must be filled in for the cancel job step operation.
18	fTruncateLogFile	1	If TRUE, the Batch Manager's log file will be truncated after printing.

BATCH COMMANDS

Batch commands allow you to

- execute a job in the primary or secondary partition with the Batch command
- create a secondary partition and install the Batch Manager in it with the Install Batch command
- display the status of batch scheduling queues, submit and remove JCL files, and cancel the current batch job with the Batch Status command (applicable in the secondary partition only)

Batch commands are described in Section 6, "Commands."

SysIn BYTE STREAM FACILITY

The SysIn byte stream facility permits keyboard byte streams to be redirected to read data embedded in the JCL file. This allows you to write programs that run in both the primary partition and in secondary partitions. In the primary partition, keyboard byte streams return data from the actual keyboard. In a secondary partition, keyboard byte streams are redirected to the SysIn byte stream, which returns data embedded in the JCL file.

The JCL file's embedded data should immediately follow the $RUN statement. After the job step finishes executing, the Batch Manager resumes processing the JCL file, beginning with the first $JCL statement following the embedded data. For example, a job step that executes a program that takes three digits using keyboard byte streams and then computes their sum would appear as follows:

```
$RUN AddThreeDigits.Run <RETURN>
1 2 3 <RETURN>
$......
```

Note that the SysIn byte stream facility only applies to keyboard byte streams. A ReadKbd operation issued by a job step executing in a secondary partition returns characters from the actual keyboard. This can cause a conflict if more than one process is reading from the keyboard at the same time.

SysOut FILE

The Batch Manager maintains a SysOut file for each batch job processed. The SysOut file can be either a disk or a spooler file. You can specify a SysOut file in the job's $JOB statement:

```
   $JOB jobname,username,[password],[SysOut] file]
```

A printer spooler file is denoted by the printer spooler queue name surrounded by brackets. Any string not surrounded by brackets is assumed to be a file specification for a disk file. All printer spooler files are deleted after printing. If no SysOut file is specified in the $JOB statement, the default SysOut file is a printer spooler file to be queued in the SPL printer spooler queue. For example:

```
    $JOB Example,Bob,,[SPLB]
```

specifies the SysOut file to be a printer spooler file queued in the SplB printer spooler queue.

```
    $JOB Example,Bob,,[Sys]<Sys>Example.SysOut
```

specifies the SysOut file to be the disk file [Sys]<Sys>Example.SysOut.

```
    $JOB Example,Bob
```

defaults the SysOut file to be a printer spooler file queued in the SPL printer spooler queue.

The Batch Manager logs in with the user profile data before opening the SysOut file, so SysOut disk files need not be full file specifications if located in the same directory as specified in the user profile file.

In the SysOut file, the Batch Manager records the time the job started and finished, echoes all JCL statements processed, and records termination information about each job step.

SysOut BYTE STREAM FACILITY

The SysOut byte stream facility permits video byte streams to be redirected to send data to a job's SysOut file. This allows you to write programs that run in both the primary partition and in secondary partitions. In the primary applications partition, video byte streams send data to the video display. In a secondary partition, video byte streams send data to the job's SysOut file.

SAMPLE FILE USING SysIn AND SysOut BYTE STREAMS

Below is a sample of JCL file that copies three bytes of embedded data to the job's SysOut file:

```
$JOB Example,sys,,[Sys]<Sys>Example.SysOut
$RUN [Sys]<Sys>CopyThreeBytes.Run
abc
$END
```

Below is the resulting SysOut file [Sys]<Sys>Example.SysOut:

```
$JOB Example,sys,,[Sys]<Sys>Example.SysOut
Execution begun:  Fri May 7, 1982 12:10 PM

$RUN [Sys]<Sys>CopyThreeBytes.RUN
abc
Termination status code:  0

$END

Execution ended:  Fri May 7, 1982 12:10 PM
```

BATCH LOG FILE

Each active Batch Manager maintains a batch log file. The batch log file is the Batch Manager's record of all the jobs it has processed, in the order processed. In this log file, the Batch Manager records the time each job started and finished, and the termination status code for the last job step executed. Each invocation of the Batch Manager appends to its existing batch log file or creates a new log file if one does not exist.

All the batch log files are located in the [!SCR]<Batch> directory. Each Batch Manager maintains a log file named xxyy.Log, where xx is the batch scheduling queue served by the Batch Manager and yy is the Batch Manager identification number. Thus, a Batch Manager with identification number ØØ serving the batch scheduling queue BATCH would maintain the log file [!SCR]<Batch>BATCHØØ.Log.

A job can send data to the log file with the $LOG statement.

The batch log file is opened long-lived by the Batch Manager so the user cannot access the file. (See the section on memory management in the CTOS Operating System Manual.) The batch log file can be printed and truncated by using the Batch Status command.

BATCH PROCESSING IN THE PRIMARY PARTITION

Batch processing in the primary partition is different from that in a secondary partition in several respects. These differences are because applications executing in the primary or interactive partition have access to the actual keyboard and screen in order to interface with the user. The differences include the following:

1. The Batch Manager processes only one batch job and then returns to the previous Exit Run file, instead of processing a queue of jobs.

2. You can cancel a job executing in the primary partition by simply pressing the ACTION-FINISH keys instead of invoking the Batch Status command.

3. Keyboard byte streams return data from the actual keyboard instead of redirecting the byte stream to read data embedded in the JCL file. That is, there is no SysIn byte stream facility in the primary partition.

4. Video byte stream output appears on the screen instead of directing this output to the job's SysOut file. The Batch Manager also echoes the processed JCL statements on the actual screen. That is, the job's SysOut output appears on the screen.

5. The Batch Manager maintains no batch log file.

As mentioned above, the job's SysOut output appears on the screen. In the primary partition of a multipartitioned operating system, this output can also be sent to a disk file or a printer spooler file, just as in the secondary partition. You specify a disk file or a printer spooler file in the $JOB statement:

```
$JOB jobname,username,[password]
[SysOut file]
```

However, the default SysOut file for a job executing in the primary partition is [Vid] byte stream rather than a printer spooler file queued in the SPL queue. The SysOut file parameter of the $JOB statement is ignored when it is executed in the primary partition of a compact operating system.

A Batch Manager executing in a primary partition skips over data embedded in a JCL file and ignores all $LOG statements. This makes it possible to submit the same JCL file for processing by a Batch Manager in either the primary or a secondary partition.

A typical sequence is given below for batch processing in the primary partition:

1. The user creates a batch JCL file using the Editor or Word Processor.

2. Using the Batch command, the user invokes the Batch Manager in the primary partition and specifies the JCL file to be processed.

3. The Batch Manager processes the JCL contained in the JCL file, loading and activating all job steps.

4. When the JCL file is completely processed, control returns to the previous Exit Run file.

JOB CONTROL LANGUAGE SYNTAX

The syntax of statements and functions in the job control language (JCL) includes metacharacters, parameters, and other variables.

METACHARACTERS

Metacharacters signify the start of a new statement or function, a continued line, a comment, a literal character, and separation of variables.

New Statement or Function

A dollar sign ($) indicates the start of a new statement. For example:

```
    $CANCEL
```

A percent sign (%) indicates the start of a new function. For example:

```
    %BREAK
```

The percent sign is also used as a parameter placeholder. (See the subsection "Parameters" below.)

CONTINUED LINE

An ampersand (&) at the end of a line indicates that the line is continued on the next line. For example,

```
    $RUN [Sys]<Sys>Example.Run,p1,p2 &
                                 ,p3,p4
```

COMMENT

A semicolon (;) indicates that the string immediately following it is a comment. Comments can be interspersed with JCL statements and can be placed anywhere on a line. For example:

```
    $LOG 'End of FileA' ;message to log file
```

Literal Character

A back slash (\) means the character immediately following it is a literal character and not a metacharacter such as

```
    \$        \&        \%        \;        \,        \\
```

For example:

```
   $RUN AddMoney.Run, \$400.00, \$200.00, \$350.25
```

Separation of Variables

A comma (,) and a space are used as separators between variables, as shown under the individual statements and functions.

QUOTED STRINGS

Any string surrounded by quotation marks is considered to be a single parameter. All characters within the quoted string are processed as literal characters. Quoted strings cannot extend across two lines.

PARAMETERS

Any number of parameters, p1, p2, ... p_n_, can be included in the CALL and RUN statements and passed to the batch run file. Parameters are separated by commas. Two adjacent commas or a comma following an empty field specify a null length parameter.

Nested parameters are enclosed in parentheses, as in the following example:

```
    $RUN FileA.Run p1,(p2,p3,p4)
```

Programs reference a nested parameter as an (i, j) pair, where _i_ is the positional index for the inner level of nesting, and _j_ is the positional index for the outer level. In the above example, p3 is referred to as (2,1). The Executive passes the command name as a parameter (0,0), which many programs depend on. You can specify the following format:

```
    $RUN (file.run,command name),P1,(P2,P3,P4)
```

(For more information on parameters, see Section 4, "Basic Concepts," and the section on parameter management in the *CTOS Operating System Manual*.)

You can pass parameters to a nested batch JCL file (the called JCL file) invoked from another batch JCL file (the calling JCL file). Passed parameters are referenced inside the called JCL file by a percent sign (%) followed by the positional index of the parameter starting at Ø.

For example, the placeholder %2 in a called batch JCL file refers to the third parameter to be passed from the calling batch JCL file.

The following example further illustrates the use of parameter placeholders. If the calling batch JCL file contains the statements

```
    $CALL Copy.JCL,InputØ.Text,OutputØ.Text
    $CALL Copy.JCL,Input1.Text,Output1.Text
    $CALL Copy.JCL,Input2.Text,Output2.Text
```

and the called batch JCL file, Copy.JCL, contains the statement

```
    $RUN Copy.Run,%Ø,%1
```

then the placeholder %1 in the called JCL file refers to the output files in the above examples.

JOB CONTROL LANGUAGE STATEMENTS

Job control language (JCL) statements direct the Batch Manager in processing batch jobs. JCL statements control load-time operations such as

- establishing the job and user names
- loading and activating run files
- passing parameters to the job step

Each JCL file can contain a single batch job.

EXAMPLE OF A JCL FILE

Figure 7-4 is an example of a format of a JCL file containing JCL statements.

```
$JOB PascalTest,Pascal,,PascalTest.SysOut
;
; This job compiles, links, and runs a Pascal program.
;
$RUN [Sys]<Sys>PascalFE.Run,PascalTest.Pas
$RUN [Sys]<Sys>Linker.Run,PascalTest.Obj,PascalTest.Run
$RUN PascalTest.Run
;
$END
```

Figure 7-4. Example of a JCL File.

Statement Descriptions

JCL statements are summarized in Table 7-4 and described in alphabetic order in the pages that follow.

Table 7-4. Summary of JCL Statements. (Page 1 of 2)

CALL	Suspends processing of the calling JCL file and starts processing the called JCL file.
CANCEL	Cancels the batch job.
CANCELONERROR	Tells the Batch Manager to cancel processing of the current batch job after any job step that terminates abnormally.
CONTINUEONERROR	Tells the Batch Manager to continue processing the current batch job even if a job step terminates abnormally.
END	Signals the end of the current JCL file.
Implicit CALL	Allows JCL statements not listed above to be used as implicit CALL statements.
JOB	Establishes a job name for the JCL file, specifies a user name for accessing user profile data, and optionally specifies a SysOut file for the job.
LOG	Writes a message to the batch log file. Not applicable in the primary partition or a compact operating system.
PATH	Requests a SetPath operation for the batch job to be executed.
PREFIX	Sets the prefix (node, volume, and directory) to be used with the implicit CALL statement.

Table 7-4. Summary of JCL Statements. (Page 2 of 2)

RETURN	Returns control to the calling JCL file from the called JCL file.
RUN	Loads and activates the specified run file.
SUFFIX	Sets the default suffix used with the implicit CALL statement.

CALL

Description

The Call statement suspends processing of the calling JCL file and starts processing the called JCL file.

CALL causes the JOB statement in the called JCL file to be ignored.

CALL allows nesting of JCL files. After the called JCL file is processed, the Batch Manager resumes processing of the original JCL file, starting with the JCL statement following the CALL statement.

Format

```
$CALL fileSpec,[parameters]
```

where

FileSpec	is a full file specification.
parameters	are the parameters to be passed to the called JCL file. (See the subsection "Job Control Language Statements.")

Example

In the following example, the JCL file Joe.JCL is called and processed, and parameters pl...p6 are passed.

```
    $CALL Joe.JCL,pl,p2,p3,p4,p5,p6
```

CANCEL

Description

The CANCEL statement cancels the batch job without returning to the calling JCL file.

CANCEL can be issued in a nested JCL file.

Format

```
$CANCEL
```

Example

```
$CANCEL
```

CANCELONERROR

Description

The CANCELONERROR statement tells the Batch Manager to cancel processing of the current batch job after any job step that terminates abnormally. Abnormal termination is defined as an ErrorExit with a nonzero parameter. (See the section on task management in the CTOS Operating System Manual.)

Processing of the current batch job remains in a cancelled state until a $CONTINUEONERROR statement is encountered. If no $CANCELONERROR or $CONTINUEONERROR statement appears in a JCL file, the default is $CANCELONERROR.

Format

```
$CANCELONERROR
```

Example

```
$CANCELONERROR
```

CONTINUEONERROR

Description

The CONTINUEONERROR statement tells the Batch Manager to continue processing the current batch job even if a job step terminates abnormally. Abnormal termination is defined as an ErrorExit with a nonzero parameter. (See the section on task management in the _CTOS Operating System Manual_.)

Processing of the current batch job continues until a $CANCELONERROR statement is encountered. If no $CONTINUEONERROR or $CANCELONERROR statement appears in a JCL file, the default is $CANCELONERROR.

Format

```
$CONTINUEONERROR
```

Example

```
$CONTINUEONERROR
```

END

Description

The END statement signals the end of the current JCL file. This statement must be placed at the end of each JCL file.

Format

```
$END
```

Example

```
$END
```

Implicit CALL

Description

The implicit CALL statement allows JCL statements not listed in this section to be used as implicit CALL statements.

The implicit CALL is a shorthand format for the CALL statement. It allows a JCL file name to be called without specifying CALL.

It causes the Batch Manager to process the JCL file {node}[Sys]<Sys>file.JCL,

where

{node}[Sys]<Sys>
is the default prefix. It is replaced by the prefix set by the PREFIX statement.

file
is the character string following the $ and terminated by a space.

.JCL
is the default suffix. It is replaced by the suffix set by the SUFFIX statement.

Format

$statement

where

statement
is a statement not listed in this section, such as COPY.

Example

In the following example, COPY is used as an implicit CALL statement. The COPY statement is equivalent to the CALL statement shown below:

```
    $COPY [Sys]<Sys>Input &
      ,[Sys]<Sys>Output

    $CALL [Sys]<Sys>Copy.JCL &
      ,[Sys]<Sys>Input &
      ,[Sys]<Sys>Output
```

With this facility, you can have JCL files such as PASCAL.JCL, COBOL.JCL, FORTRAN.JCL, or LINK.JCL in [Sys]<Sys>. The file can contain

```
$PASCAL,myprogram1
$PASCAL,myprogram2
$LINK,(my program1.obj,myprogram2.obj) &
                          ,myprogram.run
```

JOB

Description

The JOB statement establishes a job name for the JCL file, specifies a user name for accessing user configuration data, and optionally specifies a SysOut file for the job. JOB must be placed as the first statement of a JCL file.

Format

```
$JOB jobname,username[,password][,SysOut file]
```

where

jobname — is the name used to reference the job. With the Batch Status command, you can specify the jobname to view the status of a job, cancel a job, or cancel a job step.

username — is used by the Batch Manager to access a user profile file. The Batch Manager establishes a default volume, directory, and file prefix, and a password with the data contained in the User Configuration file. The user profile file used is [!Sys]<Sys>xxx.user where xxx is the username specified in the JOB statement.

In a secondary partition, if no username is specified, the User Configuration file defaults to [!Sys]<Sys>.user. In the primary partition, if no username is specified, no User Configuration file is accessed.

password — is the password used to access the User Configuration file. The password contained in the User Configuration file becomes the logged-in password, not the password in the JOB password.

SysOut file — is the destination for the job's SysOut output. The SysOut file can be a disk file or a printer spooler file. If the SysOut file is not specified, the default is a printer spooler file queued in the SPL printer spooler queue. The user can specify a printer spooler file to be queued in another printer spooler queue by making the SysOut file parameter the printer spooler queue name surrounded by brackets. Any SysOut file parameter not surrounded by brackets is considered to be the file specification for a disk file. Disk files are not spooled.

Examples

```
$JOB BudgetAnalysis,AcctDept,AcctDeptPswd &
                                ,[anotherSplQ]

$JOB Compile,Tom,TomsPswd,[Sys]<Sys> &
                         ,Diskfile.SysOut
```

LOG

Description

The LOG statement writes a message to the batch log file.

LOG is ignored in the primary partition or a compact system.

Format

```
$LOG 'message'
```

where

message is an ASCII string enclosed in single quotation marks

Example

```
$LOG 'Compiling and linking budget analysis program'
```

PATH

Description

The PATH statement requests a SetPath operation for the batch job to be executed. (See the section on file management in the <u>CTOS Operating System Manual</u>.)

Format

```
$PATH node,vol,dir,prefix,password
```

where

node	is the node name
vol	is the volume name
dir	is the directory name
prefix	is the file prefix
password	is the user's password

Example

In the following example, the path is set as shown:

```
    $PATH ,Winl,Fred,Account>,MyPassword
```

PREFIX

Description

The PREFIX statement sets the prefix (node, volume, and directory) to be used with the implicit CALL statement.

If no PREFIX statement appears in the JCL file, the default prefix is [Sys]<Sys>.

Format

```
$PREFIX {Node}[Vol]<Dir>
```

Example

```
$PREFIX {NY}[Win1]<Fred>
```

RETURN

Description

The RETURN statement returns control to the calling JCL file from the called JCL file.

When RETURN is issued in a called JCL file, the Batch Manager returns to the calling JCL file and resumes processing at the JCL statement following the CALL statement. RETURN can be conditionally executed. (See the "Job Control Language Functions" subsection.)

Format

```
$RETURN
```

Example

```
$RETURN
```

RUN

Description

The RUN statement loads and activates the specified Run file.

You can pass program parameters to the Run file. (See the "Job Control Language Syntax" subsection.)

Format

```
$RUN {Node}[Vol]<Dir>File.Run[,parameters]
```

or

```
$RUN ({Node}[Vol]<Dir>File.Run,command name) &
                                   [,parameters]
```

Example

In Example 1, FileA.Run is loaded and activated, and ParameterA is passed to it.

```
    $RUN {NY}[Win1]<Fred>FileA.Run,ParameterA
```

In Example 2, COBOL is passed as parameter (0,0) to invoke the COBOL compiler to compile ProgramA.

```
    $RUN ([Sys]<Sys>COBOL.Run,COBOL),ProgramA
```

SUFFIX

Description

The SUFFIX statement sets the default suffix to be used with the implicit CALL statement.

If no SUFFIX statement appears in the JCL file, the default is .JCL.

Format

```
$SUFFIX suffix
```

where

suffix — is the suffix to be used with the implicit CALL statement.

Example

In the following example, .BatchJCL is set as the default suffix for the implicit CALL statement:

```
    $SUFFIX.BatchJCL
```

APPENDIX A: STATUS MESSAGES

The Status Codes Manual lists all error messages generated outside the Executive. The messages generated by the Executive itself are self-explanatory and are not listed.

APPENDIX B: EXAMPLES

A SUBMIT FILE THAT USES NONCONDITIONAL EXPANSION

The Submit file shown in this example can be created using the Editor or the Record command. If the file is created using the Record command, the RETURN and GO symbols are inserted for you. If you use the Editor, you must insert the hexadecimal code for GO (1Bh) using the Editor CODE-I feature.

This example shows a Submit file that uses nonconditional expansion to allow you to pass parameters to the file at the time it is invoked. This file can be used to clean up extraneous files and to perform a daily selective backup. It allows you to vary the file specification for the file(s) to be backed up.

The following characters are the contents of a Submit file called DailyBackup.sub:

```
    Delete
    <*>*-New <*>*-Old <*>*.tmp GO
    Selective Backup
    %Ø GO
```

To use this Submit file to do a daily cleanup and selective backup of all files in your default directory, give the Submit command as follows:

```
Submit
  File List                 DailyBackup.sub
  [Parameters]              *
  [Force expansion?]
  [Show expansion?]
```

To back up all files on all directories in the same manner as above, give the Submit command as follows:

```
Submit
  File List                 DailyBackup.sub
  [Parameters]              <*>*
  [Force expansion?]
  [Show expansion?]
```

Note that the escape sequence %Ø is expanded to all files in the default directory the first way the Submit command is given above, and to all files in all directories in the second.

CREATING A COMMAND TO INVOKE A SUBMIT FILE

A Submit command can be created with the New Command command to invoke a Submit file. (See the subsection "Adding a New Command" in Section 5, "Advanced Concepts.")

EXAMPLE 1

The following example shows how to create the command Daily to be used regularly to invoke a Submit file that deletes extraneous files and performs a selective backup for your default directory.

1. Create a Submit file on the System directory with the file specification [Sys]<Sys>Daily.Sub and the contents:

   ```
   Delete
   <*>*-New <*>*-Old <*>*.tmp GO
   Selective Backup
   * GO
   ```

   where GO denotes the single character for the hexadecimal code 1Bh generated by the GO key.

2. Invoke the New Command command and create the command Daily by completing the New Command form as follows:

   ```
   New Command
     Command name              Daily
     Run file                  [Sys]<Sys>SubCmd.Run
     Field names               ____________________
     Description               'Cleanup and backup'
     [Overwrite ok?]           ____________________
     [Case (default 'ØØ')]     ____________________
     [Command file]            ____________________
   ```

   This creates the command Daily with no parameter fields.

3. To invoke the command Daily, type Daily into the command form and press GO. Extraneous files are deleted and Selective Backup is started.

EXAMPLE 2

The following example creates a Submit command to invoke a Submit file that assembles a list of files. This example also shows how to include comments in a Submit file.

First, create the temporary file Tmp.Args with the text (including quotation marks):

```
    'File list'   '[Errors only?]'  '[GenOnly,
    NoGen, or Gen]'
```

Next, invoke New Command to create the subcommand.

```
New Command
  Command name          'Assemble All'
  Run file              [Sys]<Sys>SubCmd.Run
  Field names           '@Tmp.Args'
  Description           'Assemble a list of files'
  [Overwrite ok?]       yes
```

Last, create the Submit file [Sys]<Sys>AssembleAll.Sub with the text below. The GO key, entered with CODE-I (value 1Bh), appears as a filled circle character, and the FINISH key (value Ø4) appears as a filled square character.

```
%'  A comment is inserted here to make the
%'  "%" character the first character in
%'  the file and force the Submit command to
%'  expand the text in this file.
%
Assemble
%Ø
%1
%2 <GO>
%
%'  Create a temporary file list and repeat if
%'  there are more parameters in the first field
%'  of the form.
%
%?3:Copy
[Kbd]
[Scr]<$>Temp.Fls yes
<GO& %>2 <FINISH>
%
Assemble All
@[Scr]<$>Temp.Fls
%1
%2 <GO>
:
```

Invoking the command as follows causes all of the files ending with .Asm in the logged-in directory to be assembled:

```
Assemble All
  File list                      *.Asm______________________
  [Errors only?]                 Yes________________________
  [GenOnly, NoGen, or Gen]       ___________________________
```

BACKING UP A WINCHESTER DISK

The following is an example of one way to back up Winchester disk Ø (DØ). Log in to the Winchester disk. If a volume password exists, supply it.

Before using Backup Volume, inititalize a sufficient number of floppy disks with IVolume. Backup Volume copies only file information, not all the sectors of the disk, to the archive file, which greatly reduces the space required for the backed-up files.

Since it is not possible to initialize additional floppy disks once Backup Volume is started, it is inconvenient to have initialized too few disks.

To initialize each floppy disk, invoke IVolume and fill out the form as follows:

```
IVolume
  Device name                           FØ
  [Device password]
  Volume name                           Archive
  [Volume password]
  [System Image (default = 384)]        Ø
  [Log file (default = 2 ]              Ø
  [Crash file (default = Ø]
  [Max. directories]                    1
  [Max. files on volume]                1Ø
  [Primary fileheaders only?]           Yes
  [Max. files in Sys Directory]         1Ø
  [Sys Directory password]
  [Write protect Sys Directory?]
  [Suppress format of medium?]
  [Surface tests]
  [Debug?]
  [Log file]
  [Extended floppy tracks?]
  [Double-sided floppy?]
  Bad spots (e.g., 5/2/1Ø1/26]
```

After IVolume initializes the medium, the following prompt appears:

```
    Initialize another volume?
    (Press GO to confirm, CANCEL to deny,
     or FINISH to return to the Executive.)
```

To initialize another floppy disk, insert it in the drive and press GO.

When enough floppy disks are initialized, invoke Backup Volume.

```
Backup Volume
  Volume or device name                 Win
  [Volume or device password]           Win
  [Incremental from (e.g.,
  Mon Jun 1 1981 8:00 pm]               _______________
  [Suppress backup?]                    _______________
  [Suppress verification?]              _______________
  [Archive file]                        _______________
  [Delete existing archive file?]       Yes
  [Log file]                            _______________
  [Display structures?]                 _______________
```

Backup Volume prompts to mount the first volume of the archive file:

```
Please mount [Archive]<Sys>.01
(Press GO to confirm, CANCEL to deny,
or FINISH to return to the Executive.)
```

Mount the first floppy disk and press GO.

After Backup Volume fills one disk, it prompts to mount the next:

```
Please mount [Archive]<Sys>.02
(Press GO to confirm, CANCEL to deny,
or FINISH to return to the Executive.)
```

Backup Volume reports the name of each file it archives. The report appears on the screen and, if specified, is written to a log file.

REINITIALIZING A WINCHESTER DISK

The Winchester disk can be reinitialized and restored from an archive file to reduce fragmentation. To do this, execute Backup Volume twice to create duplicate archive files, bootstrap the operating system from a system floppy disk (since IVolume destroys all files on the Winchester disk), invoke IVolume, and fill in the form as follows:

```
IVolume
  Device name                          DØ
  [Device password]                    DØ
  Volume name                          Win
  [Volume password]                    Win
  [System Image (default = 384)]       ____________
  [Log file (default = 2)]             ____________
  [Crash file (default = Ø)]           ____________
  [Max. directories]                   ____________
  [Max. files on volume]               ____________
  [Primary file headers only?]         ____________
  [Max. files in Sys Directory]        ____________
  [Sys Directory password]             ____________
  [Write protect Sys Directory?]       ____________
  [Suppress format of medium?]         ____________
  [Surface tests]                      ____________
  [Debug?]                             ____________
  [Log file]                           ____________
  [Extended floppy tracks?]            ____________
  [Single-sided mini-floppy?]          ____________
  [Bad spots]                          ____________
```

After the Winchester disk is initialized, log in to the disk with the Path form:

```
Path
  [Volume]      Win
  [Directory]   Win
  [Prefix]      ____________
  [Password]    Win
  [Node]        ____________
```

Invoke Restore and leave all the fields blank.

Restore prompts to mount the first archive file:

```
Please mount [Archive]<Sys>.Ø1
(Press GO to confirm, CANCEL to deny,
or FINISH to return to the Executive.)
```

Mount the first archive file and press GO.

Restore creates all required directories and restores each file with the specifications it had when backed up. As each volume of [Archive]<Sys> is restored, Restore prompts to mount the next volume:

```
Please mount [Archive]<Sys>.Ø2
(Press GO to confirm, CANCEL to deny,
or FINISH to return to the Executive.)
```

Restore reports each file that it restores on the screen and to a log file if specified.

After the Restore operation, all files are optimally arranged, and storage is not fragmented.

The operating system can now be bootstrapped from the Winchester disk.

APPENDIX C: PROGRAMMER'S NOTES ON THE SPOOLER

The spooler is discussed in detail in the subsection "Printer Management" in Section 5, "Advanced Concepts." This appendix contains material of interest to programmers writing application systems that use the spooler.

SPOOLER BYTE STREAMS

Spooled printing can be accessed through spooler byte streams (described in the section on the Sequential Access Method in the CTOS Operating System Manual).

The queue name must be enclosed in brackets (for example, [Joe]) to distinguish it from a file specification. The name must not match a built-in byte stream device.

During the OpenByteStream operation, the spooler byte stream creates a temporary file whose name is automatically generated to ensure its uniqueness. A WriteBsRecord or WriteByte operation transfers text to the temporary disk file and expands the disk file as necessary. The CloseByteStream operation closes the disk file, then sends the file to the specified scheduling queue. (The OpenByteStream, WriteBsRecord, WriteByte, and CloseByteStream operations are described in the section on the Sequential Access Method in the CTOS Operating System Manual.)

In a cluster environment, the temporary file is created at the master workstation. At system build time, the system administrator can specify the volume on which the temporary files are to be created. The byte stream specifies [!Scr] as the volume name of the temporary file, and the CTOS file system replaces SPL with the volume name specified at system build. The symbol "!" is interpreted by the cluster local file system to specify a volume at the master workstation. (See the section on file management in the CTOS Operating System Manual.)

In addition to the queue name, the user can optionally specify a document name following the queue name ([Joe]Report). The spooler byte stream creates the temporary file called

[!Scr]<Spl>Report$$xxxxx, where $$xxxxx is a unique sequence generated by the byte stream. The installation procedure creates directory <Spl> on the volume [!Scr].

The spooler byte stream creates temporary files without password protection and deletes them after printing. If the security of temporary files is a concern, the user should create a disk file with the byte stream, then spool it with the Print command, using the security mode (see above).

Spooled printing can be reconfigured through the ConfigureSpooler operation. (See the section on spooler management in the _CTOS Operating System Manual_.)

SPOOLER CONFIGURATION FILE REQUIREMENTS

During installation, the spooler reads a Spooler Configuration file designated by the user.

When the spooler is installed, the Spooler Configuration file must contain at least the predefined code of each printer channel to be controlled by the spooler (even if the printer is not configured at this time). The code(s) tell the spooler how much memory space to allocate for printers.

Additional information required for each printer can be specified by either

- including it in the Spooler Configuration file when the printer spooler is installed, or

- using the ConfigureSpooler operation during execution or the Spooler Status command (The ConfigureSpooler operation is described in the section on spooler managment in the _CTOS Operating System Manual_.)

The additional information required for each printer is the following:

- the name of the printer

- the name of the scheduling queue

- the Printer Configuration file specification

- the priority of the printer control process (See "Control Queue" below.)

- a code to indicate whether a banner page is printed between files

PRINTER SPOOLER ESCAPE SEQUENCES

Printer spooler escape sequences are special character sequences embedded in text files to be printed by the printer spooler. They either cause an intentional manual intervention condition when processed by the printer spooler or override the page count generated by the printer spooler. The format for a printer spooler escape sequence is

ØFFh, type, cbText, text

where

type — identifies the reason a manual intervention is required:

1 = forms change
2 = print wheel change
3 = generic printer pause
4 = page number overwrite

cbText — is the count of bytes in the following text. The maximum is 12 for types 1 and 2, and 6Ø for type 3.

text — is a character string that identifies the desired form or print wheel, the reason for the generic printer pause, or the page number.

SCHEDULING QUEUES

If an application system is reading through a scheduling queue with the ReadNextQueueEntry operation (described in the section on Queue Management in the *CTOS Operating System Manual*), it can determine if an entry is currently being printed by looking at the serverUserNum field of the returned Queue Status Block. If the serverUserNum field is ØFFFFh, the entry is waiting to be printed; otherwise, the entry is currently printing.

Table C-1. Format of a Scheduling Queue Entry. (Page 1 of 3)

Offset	Field	Size (bytes)	Description
Ø	fDelAftPrt	1	If set to TRUE (ØFFH), the spooled file will be deleted after it is printed.
1	sbFileSpec	92	The name of the file to be printed (the first byte of an "sb" string is the length of the string).
93	sbFormName	13	The name of the form to be used. If the length is zero, the standard form will be used.
1Ø6	sbWheelName	13	The name of the print wheel to be used. If the length is zero, the standard print wheel will be used.
119	cCopies	2	The number of copies of the file that are to be printed.

Table C-1. Format of a Scheduling Queue Entry. (Page 2 of 3)

Offset	Field	Size (bytes)	Description
121	bPrintMode	1	The printing mode, where Ø = normal 1 = binary 2 = image
122	fAlignForms	1	If this is TRUE, the forms alignment option will be used.
123	fSecurityMode	1	If this is TRUE, the file will be printed in security mode.
124	reserved	5	Reserved for use by the Word Processor.
129	sbDocName	92	The name of the document being printed. This is different from sbFileSpec, which is typically a temporary file in the [!Scr]<Spl> directory.
221	sbUserName	31	The client's user name.

Table C-1. Format of a Scheduling Queue Entry. (Page 3 of 3)

Offset	Field	Size (bytes)	Description
252	reserved	4	Reserved for use by the Word Processor.
256	timeQueued	4	The date/time that the print was queued.
26Ø	fSupressNewPage	1	If TRUE, the Spooler Manager will not print a form-feed at the start of the print.
261	fWPPaging	1	If TRUE, the Spooler Manager will use WP page escape sequences to determine page numbers.
262	fSupressBanner	1	If TRUE, the Spooler Manager will not print a banner or the notice file.
263	fSingleSheet	1	If TRUE, the printer attached is manual feed.
264	reserved	2Ø	Reserved for future expansion.

SPOOLER STATUS QUEUE

If an application system is reading through a status queue with the ReadNextQueueEntry operation, the only active (valid) printer status entries are those that are marked. An application system can determine which entries are marked by looking at the serverUserNum field of the returned Queue Status Block. If the serverUserNum field is ØFFFFFh, the entry is not an active printer.

Table C-2 shows the format of a status queue entry.

Table C-2. Format of a Status Queue Entry. (Page 1 of 3)

Offset	Field	Size (bytes)	Description
Ø	sbPrinterName	13	The name of the printer.
13	sbCurrentPage	13	Character sequence that defines a page number in a Word Processor print file.
26	reserved	25	
51	sbQueueName	51	The name of the queue the printer is serving.
1Ø2	bChannelNum	1	The channel used, where Ø (3Øh) = parallel port, A (41h) = serial channel A, B (42h) means serial channel B.

Table C-2. Format of a Status Queue Entry. (Page 2 of 3)

Offset	Field	Size (bytes)	Description
1Ø3	sbConfigFile	79	The name of the Printer Configuration file.
182	fAtMaster	1	TRUE if server is located at the master of a cluster.
183	bStatus	1	The printer status, where Ø = idle 1 = paused 2 = printing 3 = offline 4 = down
184	sbSpooledFile	79	The name of the currently printing file.
263	sbWheelName	13	The name of the current print wheel. If the length is zero, the standard print wheel is used.
276	sbFormName	13	The name of the current forms. If the length is zero, then the standard forms are being used.

Table C-2. Format of a Status Queue Entry.
(Page 3 of 3)

Offset	Field	Size (bytes)	Description
289	sbPauseMessage	61	The pause message to be displayed.
35Ø	fNeedWheelChange	1	TRUE if a different print wheel is needed.
351	fNeedFormsChange	1	TRUE if a different form is needed.
352	fShowPauseMsg	1	TRUE if the pause message should be displayed.
353	wsNum	2	The workstation number.
355	reserved	2	
357	sbDocName	79	The name of the document being printed.
436	sbUserName	31	The client's name.
467	timeStarted	4	The date/time that the print was started.

CONTROL QUEUE

Table C-3. Format of a Control Queue Entry.

Offset	Field	Size (bytes)	Description
Ø	bCommand	1	The command to the spooler, where Ø = halt/pause printer 1 = cancel print 2 = restart printer 3 = align forms
1	restartPage	2	The page number from which to restart printing. If this value is Ø, then the printing restarts at the beginning of the current page. If this value is ØFFFFh, then the printing starts at the next character in the file.

Table D-4 shows the keyboard codes used for most keyboard characters you might wish to include in a Submit file. For a complete list of the hexadecimal character codes for the standard character set, see Appendix B of the CTOS Operating System Manual.

Table D-1. Keyboard Codes for Use with Submit Files.

Keyboard Code (hexa-decimal)	Screen Character	Key
Ø1	up arrow	up arrow
Ø3	¢	SHIFT-6 (cent sign)
Ø4	filled square	FINISH
Ø5	empty square	PREV PAGE
Ø7	bell	CANCEL
Ø9	tab	TAB
ØA	new line	RETURN NEXT
ØB	down arrow	down arrow
ØC	form feed	NEXT PAGE
ØE	left arrow	left arrow
ØF	double dagger	MOVE
12	right arrow	right arrow
11	dagger	SCROLL UP
13	trough	SCROLL DOWN
14	raised dot	COPY
1B	filled circle	GO
2Ø	space	space
25	%	SHIFT-5 (percent sign)

APPENDIX E: SYSTEM INITIALIZATION

After the CTOS operating system finishes its initialization, it automaticlly chains to a user-specified run file. This file is specified as the "chain file" parameter in the System Build Configuration file (SysGen.Asm), and defaults to [Sys]<Sys>SysInit.Run if not specified.

SysInit runs a batch stream after a workstation is bootstrapped but before the user signs on. This batch stream can be used to initialize system services (for example, the Queue Manager, the spooler, or ISAM).

On a workstation bootstrapped from a local disk, the file [Sys]<Sys>SysInit.JCL is used as a batch stream. If this file does not exist, then SysInit exits. If there is a batch stream to be run, SysInit initializes the screen display and, if applicable, displays the most recent status. It also displays a text file if so desired. SysInit then chains to [Sys]<Sys>Batch.Run to run the batch stream. Detailed discussions of these functions follow.

SCREEN DISPLAY

If the workstation is an IWS workstation, the font file [Sys]<Sys>Sys.Font is loaded, and the screen display is initialized with 8Ø columns and 34 lines. If the workstation is an AWS workstation, the screen display is initialized with 8Ø columns and 28 lines. If the workstation is an NGEN workstation, the font file [Sys]<Sys>T1Sys.font is loaded, and the screen display is initialized with 8Ø columns and 29 lines. Errors that occur during this phase of the initialization cause the operating system to terminate.

SYSTEM FAILURE STATUS

If the operating system is rebootstrapped following a system failure, SysInit displays the system failure on the screen.

TEXT FILE

SysInit looks for a text file with the name [Sys]<Sys>SysInit.Txt. If it exists, it appears on the screen. Note that this is a convenient mechanism for broadcasting to several workstations. For example, consider a cluster configuration in which all workstations bootstrap from communcations lines. When these workstations bootstrap, any messages contained in SysInit.Txt are displayed.

BATCH STREAM

SysInit then chains to [Sys]<Sys>Batch.Run to run the JCL file for the workstation. If the chain operation fails (for example, because [Sys]<Sys>Batch.Run does not exist), SysInit error exits with an appropriate status code.

When the JCL file is processed, Batch exits.

EXAMPLE 1

A typical SysInit JCL file for a master workstation installs sytem services such as the Queue Manager and the printer spooler. For this, the user creates a file <Sys>SysInit.JCL on the [Sys] volume of the master workstation. The following JCL file installs the Queue Manager and the printer spooler using default parameters.

```
$JOB SysInit
$Run [Sys]<Sys>InstallQmgr.Run
$Run [Sys]<Sys>InstallSpl.Run
$END
```

EXAMPLE 2

A cluster has several IWS workstations without local file sytems, two of which run the printer spooler. The two special workstations are set up as type 8 workstations, by setting an internal switch. (See the "Mini-Cluster Architecture" section of the <u>System Programmer's Guide</u>.) The following batch JCL file is stored in [Sys]<Sys>WS880>SysInit.JCL on the master, and is run by the special workstations whenever they bootstrap. It installs the printer spooler (with default parameters).

```
$JOB SysInit8
$Run [Sys]<Sys>InstallSpl.Run
$End
```

GLOSSARY

Allocation Bit Map. The Allocation Bit Map controls the assignment of disk sectors. It has one bit for every sector on the disk, and the bit is set if the sector is available. The Allocation Bit Map is on disk.

Application system. An application system is any application program, written by either the user or vendor.

Archive file. Backup Volume, Selective Backup, and Restore use a multivolume data set generically referred to as an archive file. An archive file consists of several files located on one or more volumes. An archive file consists of 1 to 99 floppy disks with the same volume name and consecutive numeric suffixes (.Ø1, .Ø2, etc.). For example, one archive file can consist of [Archive]<Sys>.Ø1, [Archive]<sys>.Ø2, etc. An archive file can also be stored on magnetic tape or another larger disk.

Buffer. A buffer is an area in memory used to store information temporarily.

Byte stream. A byte stream, a concept of the Sequential Access Method, is a readable (input) or writable (output) sequence of 8-bit bytes. An input byte stream can be read until either the reader chooses to stop reading or it receives status code 1 ("End-of File"). An output byte stream can be written until the writer chooses to stop writing.

Cluster. A cluster configuration is a local resource-sharing network consisting of a master workstation and up to 16 cluster workstations.

Command file. The Command file is a file on the System directory (<Sys>) that defines the commands the Executive references. One can add information to the Command file using the New Command command, and remove it using the Remove Command command.

Command form. An Executive command form consists of the command name, parameter prompts, and parameter fields. A command form is used to supply the Executive with the information it needs to execute each command.

Command interpreter. A command interpreter is a program that allows interaction between the user and the operating system.

Command prompt. A command prompt is displayed by the Executive when it is not displaying a particular command form or executing a command. The command prompt is used to give a command to the Executive.

Default value. A default value is a value that is assumed when no other value is specified.

Directory. A directory is a collection of related files on one volume. A volume is divided into one or more directories.

Direct printing. Direct printing prints a file directly from the workstation to a printer attached to that workstation. The printer must be unoccupied before any direct printing job can be started.

Escape sequence. An escape sequence is a special sequence of characters that invokes special functions.

File. A file is a set of related records treated as a unit. For example, a file can contain a series of executable instructions, a series of data items, or a document.

File Area Block. A File Area Block (FAB) exists for each disk extent in an open file. The FAB specifies the location and number of sectors in a disk extent. The FAB is pointed to by a File Control Block or another FAB. The FAB resides in memory.

File Header Block. Each file has a File Header Block. The FHB of each file contains information about that file such as its name, password, protection level, the date/time it was created, the date/time is was last modified, the disk address, and the size of each of its disk extents. The FHB is resident on disk, occupying one sector.

File ,specification. A file specification is used in the Executive to refer to a file. The file specification is the logical name for the file and indicates its location. File specifications can be given in their full form or can be abbreviated.

Network. The network provides access to the system services of interconnected cluster configurations and standalone workstations. It uses high-speed local data lines, leased telephone lines, and packet-switched networks.

Node. A node is a location in a network of workstations. It can be a master or a standalone workstation. More than one cluster can be located at each node.

Parameter. A parameter is a string of characters representing an arbitrary value, such as a file name, a device name, a password, a "yes", or a "no". Parameters are the items of information supplied by the user when completing an Executive command form.

Parameter field. A parameter field is the highlighted area to the right of each parameter prompt in a command form. The highlight appears as you move the cursor into each parameter field.

Password. A password is a string of characters assigned to a file, directory, or volume. A valid password must be supplied before access to a file is allowed.

Protection level. File protection level controls which types of passwords are required, if any, to gain read or modify access to a specific file. Protection levels are assigned to files only. Although a directory can have a default protection level, that protection level is used to assign a protection level to each file at the time it is created. The protection level is a number used by the operating system. Protection level numbers are based on bit numbers and are not hierarchical.

Sequential Access Method. The Sequential Access Method provides device-independent access to devices (such as the screen, printer, files, and keyboard) by emulating a conceptual, sequential character-oriented device known as a byte stream.

Spooled printing. Spooled printing uses a queue system to transmit the contents of a file to the printer. With spooled printing, workstations in a cluster can print to a locally attached printer or to printers attached to other workstations in the cluster system. When a spooler has been installed at a workstation for a printer, that printer is no longer available for direct printing.

Spooler. The spooler is a dynamically installed system service that transfers text from disk files to the printer interfaces of the workstation on which the spooler is installed. It can simultaneously control the operation of several printers. A disk-based priority-ordered queue controlled by the Queue Manager contains the file specifications of the files to be printed and the parameters controlling the printing. This allows the spooler to resume printing automatically when it is reinstalled following a CTOS reload.

Submit file. A Submit file is a file that contains a sequence of characters that can be read directly from the file and processed as if the characters had been typed in from the keyboard.

Subparameter. A subparameter is one of several items specified in a parameter field.

Volume. A volume is a disk that has been formatted and initialized to store files. It may be a hard disk, floppy disk, or magnetic tape.

Volume Home Block (VHB). Each volume contains a Volume Home Block (VHB). The VHB is the root structure of information on a disk volume. The VHB contains information about the volume, such as its name and the date it was created. The VHB also contains pointers to the Log file, the System Image, the Crash Dump Area, the Allocation Bit Map, the Master File Directory, and the File Header Blocks. See the Glossary in the CTOS Operating System Manual for definitions of these structures.

Wild card character. A file specification can include a special character called a wild card character. This character tells the Executive that it should search for file specifications that match all characters given in the file specification except the wild card character(s). Where a wild card character appears, all matches are considered valid. The Executive recognizes two wild card characters: the asterisk (*), which specifies all matches of zero or more characters, and the question mark (?), which specifies all matches of exactly one character.

USER'S COMMENT SHEET

Executive Manual, 3rd Edition
A-09-00330-01-A DA-500

We welcome your comments and suggestions. They help us improve our manuals. Please give specific page and paragraph references whenever possible.

Does this manual provide the information you need? Is it at the right level? What other types of manuals are needed?

Is this manual written clearly? What is unclear?

Is the format of this manual convenient in arrangement, in size?

Is this manual accurate? What is inaccurate?

Name ______________________ Date ____________

Title ______________________ Phone ____________

Company Name/Department ______________________

Address ______________________

City ____________ State ________ Zip Code ________

Thank you. All comments become the property of Convergent Technologies, Inc.

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT #1309 SANTA CLARA, CA

POSTAGE WILL BE PAID BY ADDRESSEE

Convergent Technologies
2500 Augustine Drive
Santa Clara, CA 95051

Attn: Technical Publications

Fold Here

Please Do Not Staple

ed

Convergent Technologies

2500 Augustine Drive
Santa Clara, CA 95051
(408) 727-8830

Printed in U.S.A.