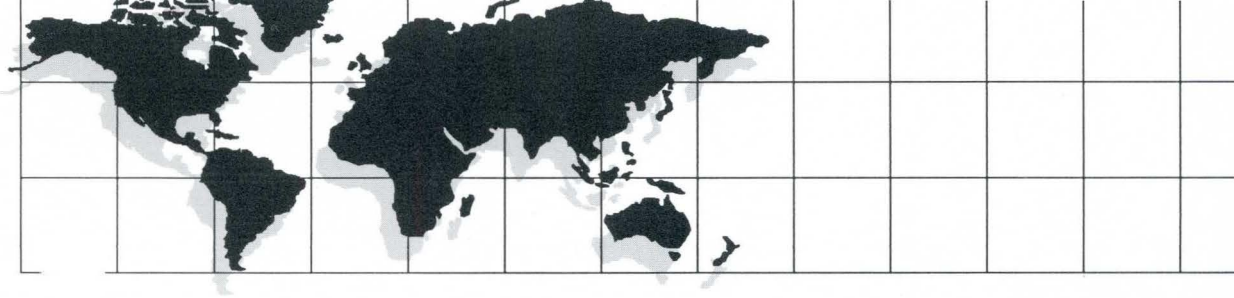


CTOS

Executive
Reference Manual

UNISYS

UNISYS

CTOS®
Executive
Reference Manual

CTOS III 1.0
CTOS II 3.4
CTOS I 3.4
CTOS/XE 3.4
Development Utilities 12.2
Standard Software 12.2
Video Access Method 4.0
Priced Item

August 1992

Printed in U S America
4357 4540–200

The names, places, and/or events used in this publication are not intended to correspond to any individual, group, or association existing, living, or otherwise. Any similarity or likeness of the names, places, and/or events with the names of any individual, living or otherwise, or that of any group or association is purely coincidental and unintentional.

NO WARRANTIES OF ANY NATURE ARE EXTENDED BY THIS DOCUMENT. Any product and related material disclosed herein are only furnished pursuant and subject to the terms and conditions of a duly executed Program Product License or Agreement to purchase or lease equipment. The only warranties made by Unisys, if any, with respect to the products described in this document are set forth in such License or Agreement. Unisys cannot accept any financial or other responsibility that may be the result of your use of the information or software material, including direct, indirect, special or consequential damages.

You should be careful to ensure that the use of this information and/or software material complies with the laws, rules, and regulations of the jurisdictions with respect to which it is used.

The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

RESTRICTED RIGHTS LEGEND. Use, reproduction, or disclosure is subject to the restrictions set forth in DFARS 252.227-7013 and FAR 52.227-14 for commercial computer software.

UNISYS

Product Information Announcement

● New Release ○ Revision ○ Update ○ New Mail Code

Title

CTOS® Executive Reference Manual

This Product Information Announcement announces the release and availability of the *CTOS Executive Reference Manual.*

This manual describes how to use the Executive command interpreter application and provides detailed information about the commands and utilities packaged with Standard Software.

Please address all technical communications relative to this document to UNISYS, Multimedia Product Information M/S 18–007, 2700 North First Street, San Jose, CA 95134–2028.

To order additional copies of this document

- United States customers call Unisys Direct at 1–800–448–1424.
- All other customers, contact your Unisys Sales Office.
- Unisys personnel use the Electronic Literature Ordering (ELO) system.

Contact your Unisys representative for pricing information.

4357 4540–200	Manual only
4357 4334–100	Manual with binder and slipcase

Announcement only:
SA, SN, SU, SW, PR5

Announcement and attachments:

System: CTOS
Release
CTOS III 1.0
CTOS II 3.4
CTOS I 3.4
CTOS/XE 3.4
Development Utilities 12.2
Standard Software 12.2
Video Access Method 4.0

Part Number: 4357 4540–200

Page Status

Page	Issue
iii	Original
iv	Blank
v through xi	Original
xii	Blank
xiii	Original
xiv	Blank
xv through xix	Original
1-1 through 1-5	Original
1-6	Blank
2-1 through 2-32	Original
3-1 through 3-440	Original
Glossary-1 through 22	Original
Index-1 through 15	Original
Index-16	Blank

Contents

About This Manual

Section 1. Using the Executive

Section 2. Features

Figures

Tables

About This Manual

Who This Manual Is For

This manual is for programmers, system administrators, and others who use the Executive extensively to perform their daily work. Although this manual is a valuable resource for complete information about the Executive, the *CTOS® Executive User's Guide* is better suited to occasional users of the Executive and those who are new to CTOS.

What This Manual Covers

The *CTOS Executive Reference Manual* documents the commands and utilities that are packaged with Standard Software 12.2. It also documents or cross references the commands that are packaged with the CTOS Development Utilities 12.2.

New Commands

The following commands have been added to this release and are documented in this manual:

Bad Spot Locater	Displays the names of files in which disk bad spots are located.
Convert Public Case	Converts public, communal, and external symbols in an object module or library to a specified capitalization format.
Create Backup Floppy	Creates a set of diskettes from which you can bootstrap a workstation.
Volume Copy	Copies the contents of one volume to another.

Changes in Functionality

The following changes to Standard Software are documented in this manual:

Disk Squash	A new field, *[Suppress confirmation?]*, has been added.
Floppy Install	A new field, *[Append to log file?]*, has been added.
Install CFA Workstation Agent	A new field, *[Maximum remote users]*, has been added.
Installation Manager	A new field, *[Append to log file?]*, has been added.
Partition Status	The display has been changed to accommodate the CTOS III virtual memory operating system.
Screen Setup	The *[Foreground color]* field can now be used to set two colors on the Executive display: one for full-intensity and one for half-intensity color.
Server Install	A new field, *[Append to log file?]*, has been added.
Tape Install	A new field, *[Append to log file?]*, has been added.
Video	The *[Foreground color]* field can now be used to set two colors on the Executive display: one for full-intensity and one for half-intensity color.

Obsolete Commands

The following commands have been obsoleted:

Backup Volume, Bind, Install, Install NLS Service, LInstall, Restore, Selective Backup, Software Install, Tape Backup Volume, Tape Restore, Tape Selective Backup, and **XESoftware Install**

Changes in Terminology

The following terms have been added or changed since the previous edition, and are used throughout this manual.

X-Bus™ workstation	Describes B26, B27, B28, B38, and B39 workstations.
X-Bus+™ workstation	Replaces the terms SuperGen® and Series 5000, which were formerly used to describe the SuperGen Series 5000 workstation.
EISA/ISA-bus workstation	Describes the SuperGen Series 3000 workstation.

Other new terms are defined in the sections to which they apply.

How This Manual Is Organized

This manual is divided into the following sections:

Section 1. Using the Executive

This section provides basic information about the Executive screen, how to issue commands, and the Executive help facility.

Section 2. Features

This section describes special features of the Executive that enhance or simplify command operations.

Section 3. Commands

This section describes the command form and parameter fields for each command. It is arranged alphabetically by command name.

In addition, a glossary and index are included near the end of the manual.

Conventions Used in This Manual

The following conventions are used throughout this manual:

- Names of forms and fields appear in italics, for example, *SignOn* form and *[Password]* field.
- File specifications, as well as volume and directory names also appear in italics, for example, *[Sys]<Sys>FormatDiskConfig.sys*.
- Variable information also appears in italics, for example,

  ...where *UserName* is the name with which the user signs on.
- Names of keys appear in small uppercase bold, for example, **GO**.
- Characters you enter appear in boldface, for example, enter **Yes**.
- Command names appear in boldface, for example, **Volume Status**.

Where to Find More Information

The following manuals and training materials are included in the CTOS documentation set:

CTOS Basic Asynchronous Terminal Emulator (ATE) User's Guide

This manual describes how to use the Basic ATE software product.

CTOS Computer-Based Training: Introduction to CTOS Systems

This computer-based training package presents online lessons and exercises to familiarize new users with basic workstation and Executive operations.

CTOS Documentation Directory

This booklet cross-references the documentation set for CTOS, Standard Software and the CTOS Development Utilities.

CTOS Editor User's Guide

This manual describes how to use the Editor application to create or modify ASCII text files.

CTOS Executive User's Guide

This manual provides step-by-step procedures for the most commonly used commands and features of the Executive.

CTOS Status Codes Reference Manual

This manual lists error codes and their meanings in numerical order.

CTOS System Administration Guide

This manual describes how to configure and maintain Standard Software on workstations and shared resource processors.

CTOS System Software Installation Planning Guide

This manual assists with the software installation process of CTOS I, CTOS II, CTOS III, CTOS/XE, Standard Software, and the Video Access Method (VAM).

You may optionally purchase the CTOS Development Utilities, which are used by programmers who develop applications for CTOS systems. The following manuals are included with the Development Utilities software:

CTOS Debugger User's Guide

CTOS/Open Programming Practices and Standards

CTOS Operating System Concepts Manual

CTOS Procedural Interface Reference Manual

CTOS Programming Guide

CTOS Programming Utilities Reference Manual: Assembler

CTOS Programming Utilities Reference Manual: Building Applications

CTOS Programming Utilities Reference Manual: Customization

CTOS Sort/Merge Programming Reference Manual

Section 1
Using the Executive

What Is the Executive?

The Executive application is the CTOS command interpreter. It associates command names with utilities and application run files. From input that you define, the Executive starts programs and passes parameters to them.

Starting the Executive

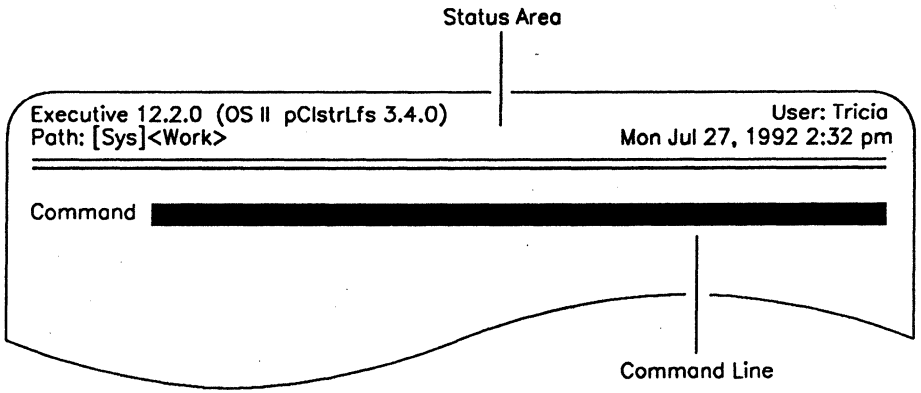
Depending on the configuration of your workstation, you can start the Executive in the following ways:

- When you sign on
- With Context Manager™
- With Presentation Manager

See the *CTOS Executive User's Guide* for information about signing on and starting applications with Context Manager. See the *CTOS Presentation Manager User's Guide* for information about starting applications with Presentation Manager.

When the Executive is ready to use, its command line appears on the screen, as shown in Figure 1-1.

Figure 1-1. Executive Screen



The status area (see Figure 1-1), displays the following information:

- Name and version number of the Executive
- Name and version number of the operating system (OS)
- SignOn user name
- Default path
- System date and time

Issuing a Command

To issue an Executive command, follow these steps:

1. On the command line, type the command name.
2. Press **RETURN** to display the command form.
3. Fill in the command form.
4. Press **GO** to execute the command.

Abbreviating Command Names

Some commands have long names that describe their functions. For most commands, however, you do not need to type the entire command name. To abbreviate a command name, type as much of it as is necessary to make it different from any other command name. For example, **Del** is an abbreviation for **Delete**.

If a command name contains more than one word, you can abbreviate each word. For example, **V S** is an abbreviation for **Volume Status**.

Command names are not case sensitive. You can type them in any combination of upper case and lower case characters, for example, **Copy**, **copy**, or **COPY**.

Command Forms

Most commands have a form containing parameter fields in which you supply information to the Executive. For example, the **Copy** command form is pictured below:

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

Parameter field names briefly describe expected parameter values. For example, *File to*, in the **Copy** command form, indicates that a file specification is expected.

In addition, certain conventions are used throughout Executive command forms to identify the following types of fields:

Mandatory	Field names that are not surrounded by square brackets must be filled in or the command cannot be executed.
[Optional]	Field names surrounded by square brackets ([ ]) are optional, depending on the results you expect from the command. When you leave an optional field blank, a default value is assigned. Default values for some fields are shown in parentheses in the command form.
[Query?]	Optional field names ending with a question mark (?) expect a **Yes** or **No** answer. If you leave a query field blank, a default value is used.

To fill in parameter fields, position the highlight and type each value. The command is not executed until you press **GO**. To move the highlight, press **RETURN**, **NEXT**, **UP ARROW**, or **DOWN ARROW**.

To enter more than one value per field, separate each value with a space, as shown in the following example:

```
Delete
 File list          File1 File2 File3
 [Confirm each?]
```

If no parameter values are required, simply press **GO** to execute a command.

Using the Help Facility

The Executive help facility provides the following information:

- Abbreviated help

  To display a list of available commands, press **HELP**.

- Full help

  To display a description of each command, press **HELP**, then **CANCEL**, then **HELP** again.

- Specific help

  To display a description for a specific command, first display the command form, then press **HELP**.

- Window-interface help

  To display help for commands and applications that use the window interface, select *Help* on the menu bar. See Section 2, "Features," for information about using the window interface.

Section 2
Features

Overview

The Executive provides the following features to enhance or simplify command operations:

Keyboard	The keyboard contains keys that perform specific functions in the Executive.
Mouse	You can use a mouse with several Executive commands. When the mouse is available, its cursor appears on the screen.
Wild-card characters	You can use wild-card characters in command forms to replace all or part of a file specification. These are useful for locating files, performing commands on groups of files, and minimizing typing in command forms.
Device specifications	With some commands, you can direct command output to a device, rather than to a file. For example, you can use the Copy command to print a file rather than to copy it to a disk file. You use a device specification to do this.
At-Files	You can use the contents of a text file, called an at-file, as a parameter value in Executive command forms. This is useful for entering lengthy parameter values that don't fit in the command form.

Passwords	You use Executive commands to assign passwords and protection levels to the file system. In addition, you may need to enter a valid password before you can execute certain commands.
Command files	To execute commands, the Executive reads information from a command file. Command files can be customized and assigned for specific purposes on a user-name basis.
Submit facility	You can create executable files with the submit facility script language. These are useful for performing repetitive tasks or for simplifying command operations.

Keyboard

A variety of keyboard models are available for CTOS workstations, including PC-style keyboards for X-Bus+ and EISA/ISA workstations.

The workstation keyboard contains a number of special-purpose keys. Combinations of keys, such as holding down **CODE** while pressing **MARK**, provide additional functions. Special-purpose keys are described in Table 2-1. Commonly used PC key names are cross referenced to CTOS key names in Table 2-2, although on most PC-style keyboards, the keys are labeled with both CTOS and PC key names.

Type-Ahead Feature

When you enter characters into Executive command fields, you may sometimes type faster than characters are displayed. This happens when the workstation is burdened with simultaneous tasks, such as printing, or is executing other applications with Context Manager.

When this happens, keystrokes are not lost. They are stored in the *type-ahead buffer*, which saves keystrokes until the Executive is free to process them. Delays of this type are infrequent and last only a few seconds.

To clear the type-ahead buffer, press **ACTION-DELETE**.

Table 2-1. Keystroke Functions

Keystroke	Function
ACTION-DELETE	Clears the type-ahead buffer (see "Type-Ahead Feature," below).
ACTION-FINISH	Terminates a command that is executing.
ARROWS	Move the cursor or highlight in the indicated direction.
ACTION-OVERTYPE	Turns off the screen. Press any key to turn the screen on again.
BACKSPACE	Deletes the character to the left of the cursor.
CANCEL	Cancels a command form and redisplays the command line, without executing the command. It is also used for special purposes, which are described on the screen, during execution of certain commands.
CODE-'	Displays the next character typed as a keyboard literal (see "Symbols and Literal Characters," below).
CODE-A to **CODE-Z**	Replays a macro that is created with **CODE-MARK** (see its description below).
CODE-BACKSPACE	Deletes characters from the left of the cursor to the preceding space.
CODE-BOUND	Executes the **Stop Record** command, when the Executive is recording; see also **CODE-MARK**, below, and the **Stop Record** command, in Section 3.
CODE-DELETE	Deletes all characters on the command line or in a command form field.
CODE-DOWN ARROW	See **CODE-UP ARROW**, below.
CODE-F2 to **CODE-F10**	Stores a filled-in command form as a function key macro. See the *Executive User's Guide* for information about creating macros.
CODE-GO	Activates the Debugger, if it is installed. (If you accidentally activate the Debugger, press **GO** to exit it.)
CODE-LEFT ARROW	Moves the cursor to the leftmost character in a command form field.

continued

Table 2-1. Keystroke Functions (cont.)

Keystroke	Function
CODE-MARK	Allows you to define **CODE-***Letter* for a macro and begins the **Record** command. See also **CODE-A** through **CODE-Z** and **CODE-BOUND**, above, and the **Record** command, in Section 3.
CODE-NEXT PAGE	Turns off the pause that occurs before command output is scrolled off the top of the screen.
CODE-RIGHT ARROW	Moves the cursor to the rightmost character in a command form field.
CODE-UP ARROW	Activates the extended redo feature, which scrolls through the command forms of previously executed commands, starting with the command most recently executed. After scrolling back through command forms, **CODE-DOWN ARROW** scrolls the other direction. See also "Redo Features," earlier in this section.
DELETE	Deletes the character at the cursor position.
F1	Displays the filled-in command form for the last command you executed. (See also **CODE-UP ARROW**, above and **SHIFT-GO**, below).
F2 to **F10**	User-definable function keys for storing macros. (See the *CTOS Executive User's Guide* for information about creating function key macros.)
FINISH	Cancels a command form and redisplays the command line, without executing the command. It is also used for special purposes, which are described on the screen, during execution of certain commands.
GO	Executes a command.
HELP	Lists the Executive commands available for the current user, when pressed while the command line is displayed. When pressed a second time, it displays a brief description of each command. When pressed while a command form is displayed, it lists the help description for that command only.

continued

Table 2-1. Keystroke Functions (cont.)

Keystroke	Function
NEXT	Moves the cursor to the next field in a command form (the same as **UP ARROW**, **RETURN**, and **TAB**).
NEXT PAGE	Continues execution of a paused command. (The Executive does not scroll output off the screen until you have a chance to read it.) The pause before scrolling can be suppressed by pressing **CODE-NEXT PAGE**.
OVERTYPE	Types new characters over existing characters, rather than inserting them; also, **BACKSPACE** moves across existing characters rather than deleting them. When this key is activated, its light turns on.
RETURN	Moves the cursor to the next field in a command form (the same as **DOWN ARROW**, **NEXT**, and **TAB**).
SCROLL UP	See **NEXT PAGE**, above.
SHIFT-GO	Executes the current command but suppresses the redo and extended redo features. When a command is executed with **SHIFT-GO**, it is not stored on **F1**, nor is it added to the stack of commands you can scroll through with **CODE-UP ARROW** (see the documentation for those keystrokes, above).
SHIFT-LEFT ARROW	Moves the cursor five characters to the left.
SHIFT-RIGHT ARROW	Moves the cursor five characters to the right.
TAB	Moves the cursor to the next field in a command form (the same as **DOWN ARROW**, **NEXT**, and **RETURN**).

Table 2-2. PC-to-CTOS Equivalent Keystrokes

CTOS Key Name	PC Key Name
BOUND	SCROLL LK
CANCEL	ESC
CODE	CTRL
COPY	ALT
DELETE	DEL
GO	SYS REQ
MARK	NUM LK
MOVE	BREAK
NEXT PAGE	PGDN
OVERTYPE	INS
PREV PAGE	PGUP
RETURN	ENTER
SCROLL DOWN	END
SCROLL UP	HOME

Redo Features

The Executive provides two redo features so that you can quickly reexecute commands you performed earlier. To redo the most recently executed command, press **F1**; the filled-in command form is displayed. You can then reexecute the command as is, by pressing **GO**, or you can make changes to the command form before pressing **GO**.

The Executive also keeps tracks of less recently executed commands. To scroll back through previously executed command forms, press **CODE-UP ARROW**. If the command form contained wild-card characters (described later in this section), press **DOWN ARROW** to fill in the form. Once you have scrolled back through command forms, press **CODE-DOWN ARROW** to scroll the other direction. To reexecute a command, press **GO**.

To suppress both redo features for a command, execute it with **SHIFT-GO**. This prevents a command form from being stored on **F1** or in the extended redo buffer.

Function Keys

You can use function keys **F2** through **F10** to store and recall completed command forms. This feature, sometimes called *function-key macros*, is particulary useful for storing commands that you use often, such as the **Files** or **Path** command for a frequently accessed volume and directory.

To store a command on a function key, complete the command form according to your needs and then press **CODE-***FUNCTION KEY* to store the command form (where *function key* is any of the function keys labeled **F2** through **F10**).

If a macro is already assigned to the function key you select, the existing macro is overwritten.

To recall a command that is stored on a function key, press the function key on which you have stored the command. The command form is displayed, with previously specified parameters filled in; those may be changed or left intact. Press **GO** to execute the command.

Function-key macros are submit files stored by user name in a directory named *[Scr]<Exec>*. For example, if you are signed on as Jim and store a command on **F9**, its submit file is named *[Scr]<Exec>JimF9.sub*.

Note: *On systems that use a memory disk for the* [Scr] *volume, the submit files for macros are stored in* [Sys]<Exec> *and then placed in* [Scr]<Exec> *on the memory disk as they are used. See the* CTOS System Administration Guide *for information about memory disks.*

Such a submit file can be edited to contain additional keystrokes or commands. For example, a GO symbol (•) can be added to make the macro execute directly from the function key, without the need to press GO on the keyboard.

Sumbit files for function-key macros cannot contain commented text. Therefore, submit files created with the **Command File Editor** cannot be executed from a function key until commented command fields and other text have been deleted with a text editor.

Symbols and Literal Characters

Many special-purpose characters, such as the trademark symbol and the English pound sign, do not appear on the keyboard. In addition, certain symbols are associated with specific keystrokes. For example, the filled square is the keyboard symbol for FINISH, and the filled circle is the keyboard symbol for GO. Characters associated with keystrokes are called *literal characters* or *keyboard literals.*

To type a literal character or a symbol, press CODE-' (single quotation mark). Nothing appears on the screen, however, the next character or keystroke combination you type is interpreted literally. For example, the GO key produces its symbol (•) rather than executing its function.

See the *CTOS Procedural Interface Reference Manual* for complete tables of keyboard literals and keystrokes for special characters.

Mouse

When using certain interactive commands, such as the **System Manager**, you can select menu items or function keys with a mouse. When the mouse is available, its pointer (usually an arrow or a rectangle) appears on the screen.

Using the Mouse

To use the mouse, position the pointer on the menu choice or function key you want to select, then click the **Mark** button. Figure 2-1 shows the **Mark** button on a three-button mouse. Its position is the same on a two-button mouse.

Setting Up a Left-Handed Mouse

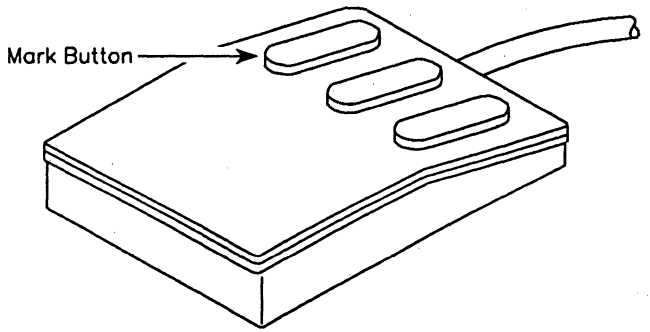
To reverse the right-to-left orientation of the mouse buttons, make the following user configuration file entry:

:LeftHanded:Yes

See the *CTOS System Administration Guide* for information about editing user configuration files.

Note: *Mouse buttons reverse only if the application with which you are using the mouse reads the user configuration file.*

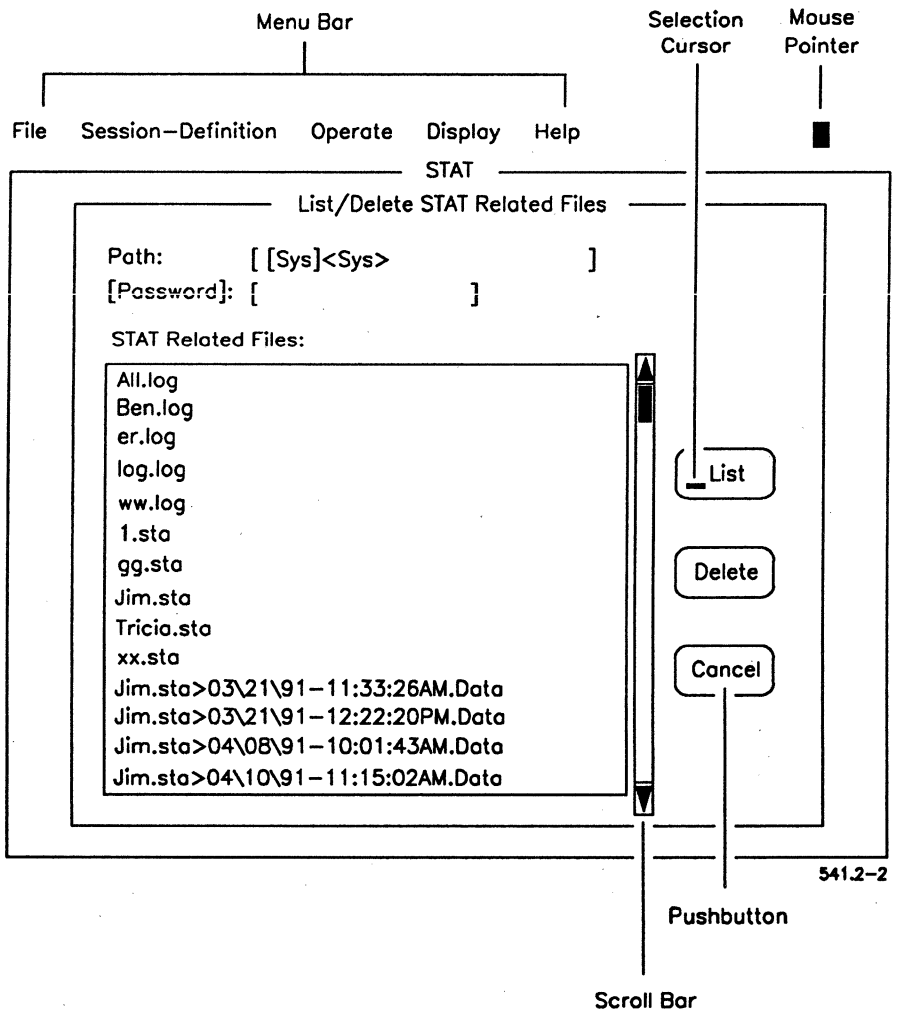
Figure 2-1. Mark Button on the Mouse



Window Interface

Some applications present a window interface, as shown in Figure 2-2. The purpose of a window interface is to provide a consistent method of use for commands and applications.

Figure 2-2. Window Interface Example Screen



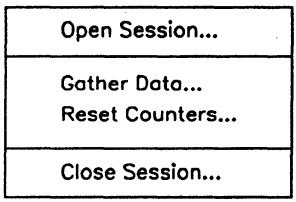
The window interface consists of the following on-screen components:

Menu bar

When you start the window interface, a menu bar appears at the top of the screen. (See Figure 2-2.)

Pull-down menus

When you select from the menu bar, a pull-down menu is displayed, from which you may select one choice. An example is shown below:



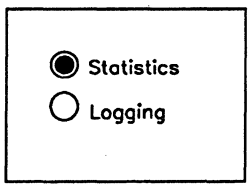
541.2-A

Pushbuttons

When applicable, pushbuttons appear within a window, as shown in Figure 2-2. You may select one pushbutton from the displayed group.

Radio buttons

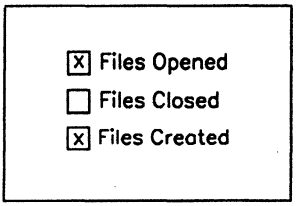
When applicable, radio buttons are displayed, as shown in the following example. You may select one radio button per group.



541.2-B

Check boxes

When applicable, check boxes appear in a small window, as shown in the following example. You may select one or more check boxes from the displayed group.



541.2-C

Selection list

A selection list is displayed in its own window, and a scroll bar, as shown in Figure 2-2, is present on the right side of that window.

Entry fields

Entry fields are predefined areas where the user types parameter values for the command.

Navigating and Making Choices

You can navigate around the window interface and make selections with either a mouse or the keyboard.

To navigate and select with a mouse:

1. Move the mouse pointer to the desired location.
2. Click **Mark**.

To navigate and select with the keyboard:

1. Use the **TAB** key to move the selection cursor to the desired location.
2. Press **SPACEBAR** to select an item.
3. Press **RETURN** to activate your selection.

Table 2-3 provides a complete list of keystrokes that allow you to navigate and make choices with the keyboard.

Table 2-3. Window Interface Keystroke Functions

Keystroke	Function
ARROWS	Move the cursor in the indicated direction. Note that when a menu bar pull-down (described above) is displayed, you can cycle through the other menu bar pull-downs by pressing **RIGHT ARROW** and **LEFT ARROW**.
COPY-*Character*	Activates a choice, such as a pushbutton. (Character corresponds to the bold character in that choice.)
F10	Activates the menu bar.
RETURN	Activates a choice.
SPACEBAR	Marks an item for selection.
TAB	Moves the selection cursor to the next field.

Scrolling

When information exceeds window borders, a scroll bar appears, as shown in Figure 2-2.

To scroll with a mouse:

1. Position the mouse pointer on the lower arrow within the scroll bar (to scroll up) or the upper arrow (to scroll down).
2. Click **Mark**.

To scroll with the keyboard:

1. Press **TAB** until the selection cursor is positioned within the scrollable window.
2. Continuously press **DOWN ARROW** (to scroll up) or **UP ARROW** (to scroll down). Scrolling begins when the cursor reaches the window border for the direction you want to scroll.

Help

The *Help* pull-down menu provides general information about using the window interface, as well as information about the specific command or application you are using. See your XVT documentation for information about using this help facility.

Wild-Card Characters

When executing commands, you can include *wild-card characters* (sometimes called metacharacters) in directory or file names. When you use a wild-card character, the Executive searches for every name that matches the characters you have typed, except for the wild cards.

The Executive recognizes the following wild-card characters:

asterisk (*)	Matches zero to all characters in a directory or file name. For example, *ab*z* matches all file names beginning with "ab" and ending with "z", regardless of how many characters are in between.
question mark (?)	Matches exactly one character in a directory or file name. For example, *ab?d* matches only those file names with exactly one character between "ab" and "d".
tilde (~)	Is an exclusionary wild card for use with the **Files** command only. It matches all characters except those that immediately follow it. See the **Files** command, in Section 3, for examples.

See the *CTOS Executive User's Guide* for detailed examples of using wild-card characters.

Invalid Uses of Wild-Card Characters

Wild-card chracters are not recognized in volume or device names.

When specified in multiple fields of a command form, wild-card characters must match in number and pattern. The following example shows a command form containing mismatched wild-card characters:

```
Rename
 File from          <*>Section.*
 File to            <Final>08339.*
 [Overwrite ok?]
 [Confirm each?]
```

If wild card entries do not match in number and pattern, the following error message is displayed:

File to has a mismatched wild card.

The **Restore Archive** command, however, presents an exception to this rule. See the description of it, in Section 3.

Wild-Card Character Expansion

For most commands, the Executive expands wild-card characters in the command form, as soon as you press **RETURN** or **GO**. Expansion means that the actual file names located with the wild-card characters are displayed in the command form. If the wild-card characters you specify match many file names, expansion can take several seconds.

For some commands, however, expansion does not appear in the command form. This improves the speed of commands, such as **Copy** and **Rename**, that are frequently executed on large groups of files.

Expansion can also be suppressed for other commands by placing an asterisk (*) in the command form field within the command file that controls the command. See the **Command File Editor**, in Section 3, for more detailed information.

Device Specifications

With the Executive, you can use device specifications in command forms to do the following:

- To specify a device, such as disk drive or a tape drive
- To print command output
- To type directly from the keyboard to a file, a printer, or the screen
- To use the "null" device, which eliminates command output

Table 2-4 lists system-defined device specifications you can use with the Executive. In most cases, device names are enclosed in square brackets ([ ]) when you use them in command forms.

Table 2-4. Device Specifications

Type of Device	Specification
Screen	*[Vid]*
Keyboard	*[Kbd]*
Printer channels	
Parallel	*[Lpt]*
Serial	*[Ptr]A* or *[Ptr]B*
Communications channels	*[Comm]X*, where *X* is the port to which the communication line is attached.
Most hard disk drives	*[dn]*, where *n* is the number of the drive, depending on its location on the workstation or shared resource processor™.*
SMD disk drives	*[sn]*, where *n* is the number of the SMD drive, depending on its location on the shared resource processor.*
Floppy disk drives	*[fn]*, where *n* is the number of the drive, depending on its location on the workstation.*
Null device	*[Nul]*
Scratch volume	*[Scr]*
SCSI devices (other than disks)	*[zn]*

*See the *CTOS System Administration Guide* for more detailed information about disk device names.

Directing Command Output to a Printer or a File

With some commands, you can direct output to a printer or to a disk file. The following example shows how to print a file with the **Copy** command; note that the device specification for the printer is used in the *File to* field.

```
Copy
  File from          [d0]<Personnel>Address.list
  File to            [Lpt]
  [Overwrite ok?]
  [Confirm each?]
```

Many commands have a *[Print file]* or a *[Log file]* field for sending command output to a file or a printer. If you specify a print file that already exists, its contents are overwritten with the current command ouput. Conversely, if you specify a log file that already exists, current output is appended to its existing contents.

The following example shows how to send output from the **Files** command to a print file:

```
Files
  [File list (*)]
  [Details?]
  [Print file]                    All.fls
  [Suppress sort?]
  [Max columns]
  [Sort by suffix?]
  [Suppress error messages?]
```

Specifying a Tape Drive

To use a tape drive for backup, restore, or copy operations, you specify a tape drive name and, optionally, a tape file mark. The tape drive name determines the tape drive that is used; the tape file mark determines where on the tape the file is written to or read from.

You can use the following tape specifications:

- *[DeviceName]*0 reads or writes data starting at the beginning of the tape.
- *[DeviceName]*+ writes data after the last tape file mark.
- *[DeviceName]n* (where *n* is the number of the tape file mark) reads data starting at the specified tape file mark.

Note that *[DeviceName]*+ is for write-to-tape operations only, while *[DeviceName]n* is for read-from-tape operations only.

Tape drive names are assigned when you install the Sequential Access Service. See the documentation for the **Install Sequential Access Service** command, in Section 3.

Typing to a File or Printer

You can send keystrokes directly to a file or a printer. To do this, you use the **Copy** command; it works differently than when you are copying files, as described below:

- After you press GO, the **Copy** command waits for your input.
- As you type, characters do not appear on the screen; instead, they are sent to the printer or file you specified.

To send keystrokes from the keyboard to a file specification, you copy from the device name *[Kbd]*, rather than from a file specification, as shown in the following example.

```
Copy
 File from          [Kbd]
 File to            [d0]<Dir>KeystrokeFile
 [Overwrite ok?]
 [Confirm each?]
```

To send keystrokes from the keyboard to a printer, you copy from the device name *[Kbd]* to the device name for the printer you want to use, as shown in the following example:

```
Copy
  File from          [Kbd]
  File to            [Queue1]
  [Overwrite ok?]
  [Confirm each?]
```

After you have finished typing to a file or a printer, press **FINISH** to terminate the **Copy** command.

Using the Null Device

You can eliminate command output by specifying the null device. This is most frequently used to test results without actually executing a command. To see how this works, you can use it with the **Copy** command. The following example shows how a list of files is copied to *[Nul]*; you can watch how the command executes without actually copying the files.

```
Copy
  File from          [Sys]<Sys>*.user
  File to            [Nul]*
  [Overwrite ok?]
  [Confirm each?]
```

At-Files

You can use the contents of an ASCII text file as a parameter value. To do so, you specify an "at" symbol (@) as the first character in the file specification, as shown in the following example:

```
LCopy
  File list                  @[f0]<Programs>All.fls
  [File prefix(s) from]
  [File prefix(s) to]        [d1]<Programs>
  [File suffix(s)]
  [Overwrite ok?]
  [Confirm each?]
  [Continue on error?]
  [Verify copy?]
```

The use of at-files is not limited to a particular type of parameter value. However, they are frequently used in the following situations:

- To specify a list of parameter values, such as file or directory names
- To enter a parameter value that is too long to be typed in a command form field

Each of these situations is described in more detail below.

Specifying a List of Parameter Values

At-files are frequently used to perform a command with a list of parameter values. You can create a list of file names, to be used as an at-file, with the **Files** command. You can create other types of lists, such as directory or command names, with a text editor.

To create a list with the Editor application, open a new file and type the parameter values (see the *CTOS Editor User's Guide* for information about using the Editor). Separate each parameter value with a space, **RETURN**, **NEXT**, or **TAB** character. If you use a word processing program, such as OFIS™ Document Designer™, you must remove formatting before using the file as an at-file (see your word processing documentation).

To create a file list with the **Files** command, enter a file specification in the *[Print file]* field of the command form, as shown in the following example.

```
Files
  [File list (*)]              *.stuff
  [Details?]
  [Print file]                 Stuff.fls
  [Suppress sort?]
  [Max columns]
  [Suffix sort?]
  [Suppress error messages?]
```

The print file, *Stuff.fls*, can then be used as an at-file by preceding it with an @ sign when you enter it in a parameter field, as shown in the following example:

```
Selective Archive
 File list                                  @Stuff.fls
 [Incremental from]
 [Confirm each?]
 [Archive dataset ([QIC])]                  [f0]<Sys>TFS1
 [Delete existing archive dataset?]
 [Print file]
```

When the Executive expands the at-file, the names of files contained in *Stuff.fls* appear in the command form, as shown in Figure 2-3.

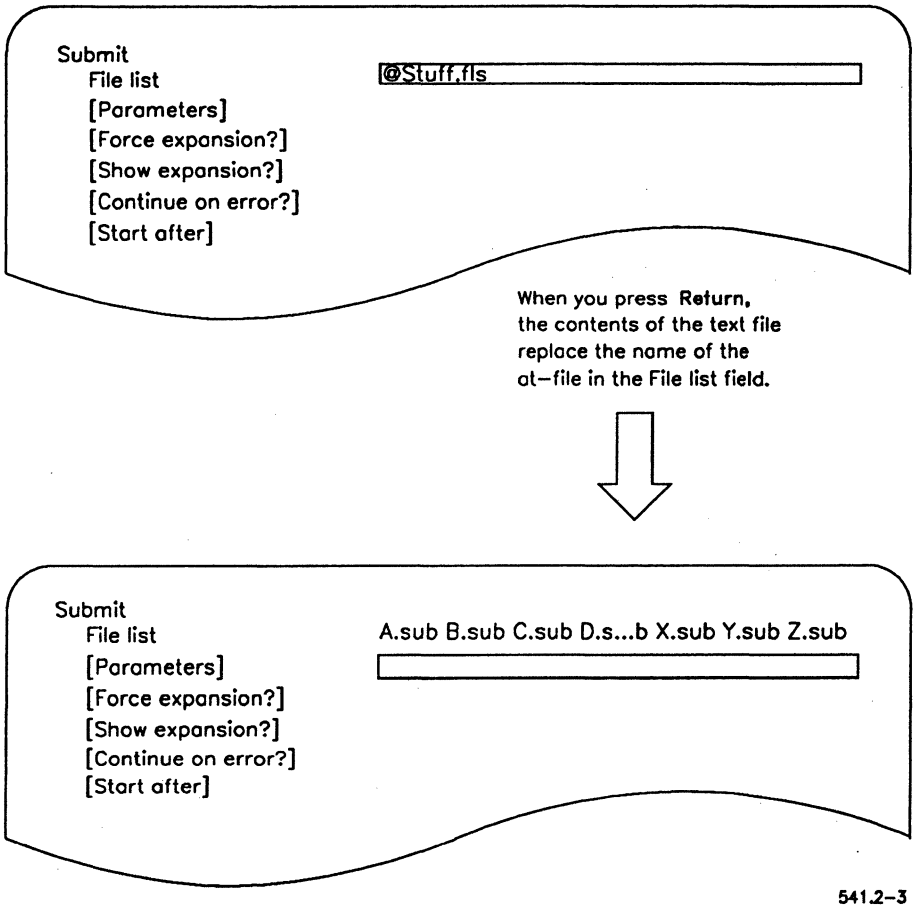
You can use the @ symbol with full or partial file specifications, but it is always the first character to be typed in the parameter field, as shown below:

@[d1]<Work>Stuff.fls

@<Work>Stuff.fls

@Stuff.fls

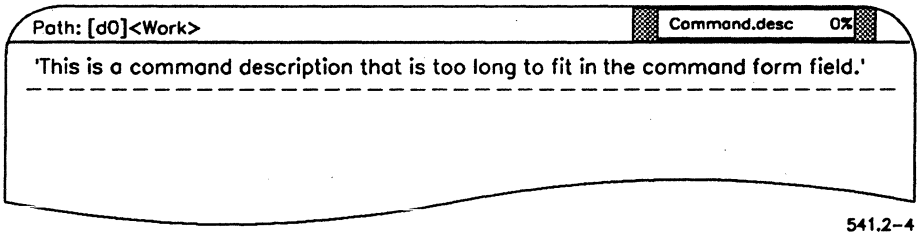
Figure 2-3. At-File in a Command Form



Entering a Lengthy Parameter Value

In some situations, parameter values are too long to type in the command form field. For example, the *[Description]* field for **New Command** often requires a sentence. In such a case, you use a text editor to create a file in which you type the description. Figure 2-4 shows a sample file named *[d0]<Work>Command.desc*, created with the Editor application.

Figure 2-4. Parameter Entered in a Text File



To use the contents of this file as a parameter value, type the file specification preceded by the @ symbol, as shown in the following example. The contents of the file are expanded in the *[Description]* field after **RETURN** or **GO** is pressed.

```
New Command
 Command name          'Sample Command'
 Run file              [Sys]<Sys>Sample.run
 [Field names]
 [Description]         @[d0]<Work>Command.desc
 [Overwrite ok?]
 [Case (00)]
 [Command file]
 [Public command?]
```

The sample command name shown above, and the description typed within the at-file, are enclosed in single quotation marks ('). Parameter values containing spaces must be enclosed in single quotation marks; otherwise, the Executive interprets each word as a separate parameter value.

File System Security

The operating system provides a multilevel approach to file system security. It allows you to assign passwords to volumes, directories, and files, and to set protection levels for individual files.

This section describes system security as it applies to the Executive. See the *CTOS System Administration Guide* for more detailed information on system security.

Device Passwords

A device password is required to format a disk or to back up a corrupted disk. Device passwords are assigned to hard disk drives by the operating system. For the prebuilt operating systems, device passwords match device names; that is, *d0*, *d1*, and so on. If you are using a customized operating system, the device passwords may be different.

Some Executive command forms call for either the volume or device password. If you are working with a valid volume, specify the volume password. If you are working with a new or corrupted volume, specify the device password.

Volume Passwords

A volume password is assigned when a disk is initialized. If a volume is password protected, the password is required to reinitialize the disk or to back it up. In addition, the volume password provides unrestricted access to any file or directory on the disk.

A volume password is the key to file system security disk. One must be assigned to enable directory and file protection levels on the volume.

See the **Format Disk** command and **Change Volume Name** command, in Section 3, for information about assigning volume passwords.

Directory Passwords

A directory password is assigned when a directory is created, or later with the **Set Directory Protection** command. If a directory is password protected, the password is required to copy, create, or rename files or to remove the directory.

See the **Create Directory** command and the **Set Directory Protection** command, in Section 3, for information about assigning directory passwords.

File Passwords

If a directory is password protected, files created in it inherit the directory password. However, unique passwords can be assigned to files with the **Set Protection** command. File passwords are most often used to restrict access to certain files.

Each file also has a protection level that determines its access level:

- *Read access* means a file can be opened but not modified. Read access is required for the operating system, configuration files, and application run files.
- *Modify access* means a file can be created or changed.

The protection level also determines what level of password is required to gain access to a file. Protection levels are described in Table 2-5.

Note that protection levels apply to files only. The default protection level you specify when you create a directory subsequently applies to all files created there. It has no bearing on the directory itself. For example, level 15 assigns unrestricted access to the files in a directory. However, if that directory is password protected, a volume or directory password is still required to create files or remove the directory.

Table 2-5. Protection and Access Levels

Protection Level	Password Level Required		Description
	To Read	To Modify	
15	None	None	Unprotected. No password is required to read or modify the file.
5	None	Volume Directory	Modify protected. No password is required to read the file. A volume or directory password is required to modify the file.
0	Volume Directory	Volume Directory	Read and modify protected. A volume or directory password is required to read or modify the file.
7	None	Volume Directory File	Modify password. No password is required to read the file. A volume,directory, or file password is required to modify the file.
3	Volume Directory File	Volume Directory File	Access password. A volume, directory, or file password is required to read or modify the file.
1	Volume Directory File	Volume Directory	Read password. The file can be read with a volume, directory, or file password, but a volume or directory password is required to modify it.
23	None	Volume File	Nondirectory modify password. No password is required to read the file, but a volume or file password is required to modify it.

continued

Table 2-5. Protection and Access Levels (cont.)

Protection Level	Password Level Required		Description
	To Read	To Modify	
19	Volume Directory File	Volume File	Nondirectory access password. The file can be read with a volume, directory, or file password, but a volume or file password is required to modify it.
51	Volume File	Volume File	Nondirectory password. A volume or file password is required to read or modify the file.

Supplying a Password

When a volume, directory, or file is password protected, a valid password must be supplied to perform certain operations. In some cases, the password you supply becomes the default or active password.

The various methods for supplying a password to the Executive are described below.

In the SignOn Form

When you enter a password in the SignOn form, it becomes the *active password* and takes effect for everything you do on the system. The active password (also called the *default password*) does not need to be resupplied when you execute a command. For example, if you sign on with the volume password, you have unrestricted access to the disk. This is convenient, but be aware that unrestricted access includes the capability to reformat disks or remove directories. You are not prompted to enter the password again before performing these operations.

From the User Configuration File

If the user configuration file contains a *:SignOnPassword:* entry, that password supersedes the password entered in the SignOn form and becomes the active password. For example, some users sign on with a file password that provides access to the user configuration file only. Then, the SignOn program reads a directory password from the user configuration file. See the *CTOS System Administration Guide* for a detailed explanation of user configuration files.

With the Path Command

You can use the **Path** command to change or disable the active password. The password you enter in the command form then takes effect for all operations you perform on the system. See Section 3, for more information about the **Path** command.

In a Command Form

Some command forms provide a *[Password]* field, which allows you to temporarily override the active password. This does not change the password setting, but is effective only while that one command is executing. The **Path** command, as described above, is the only exception.

As Part of a File Specification

In some cases, you might need to use two different passwords to execute a command, for example, if you are copying a file from one disk to another. In this case, you can append a password to a file specification by preceding the password with a circumflex (^), as shown in the following example.

```
Copy
  File from          StatusReport
  File to            [!d1]<TS>StatusReport^####
  [Overwrite ok?]
  Confirm each?]
```

This method overrides the active password for the field in which it is entered.

Command Files

To execute a command, the Executive reads the following information from a command file:

- Command name
- Command form
- Command case
- Run file name
- Executive Help description

Private and Public Command Files

Each user has access to both a private and public command file, as described below.

A *private command file* contains commands that are available on a user name basis. For example, Joe can use a different command file than Bob. The default private command file is *[Sys]<Sys>Sys.cmds*. A customized private command file can be assigned with the *:ExecCmdFile:* user file entry. (See the *CTOS System Administration Guide* for information about user configuration files.)

A *public command* file is stored on the server and is shared by all users on the cluster; it cannot be customized for individual users. The public command file is named *[!Sys]<Sys> Cluster.cmds*. It contains commands that are created during public software installations.

When you initiate a command, the Executive first reads the private command file. If the command name is not contained in the private command file, the Executive reads the public command file.

Editing Command Files

The **Command File Editor** command is used to create and customize command files. Customized command files may contain any subset of the commands available on the system. Frequently, customized command files contain only the commands needed for particular job duties. Customized command files are often created when several users share a workstation or when a workstation boots from the server, rather than from a local disk.

See the **Command File Editor** command, in Section 3, for more information about command files.

Submit Facility

With the Executive submit facility, you can create executable files using Executive commands. The submit facility reads keystrokes from a file, just as if they were typed on the keyboard. Submit files can range from macros, which carry out commands with a single keystroke, to sophisticated routines that function as programs.

Simple submit techniques are described for the **Record** command in Section 3, and in the *CTOS Executive User's Guide*. See the **Submit** command, in Section 3, for detailed information about advanced techniques for writing submit files and using the submit facility.

Section 3
Commands

Standard Software Packages

This section contains detailed information about the Executive commands included with Standard Software. Commands are arranged in alphabetical order by name.

Standard Software contains many commands, which are divided into packages according to functionality. When you install Standard Software, you selectively install only the command packages you need to use.

In this manual, commands are identified by the following categories, which correspond to Standard Software packaging. For more information about Standard Software installation and packaging, see the *CTOS System Software Installation Planning Guide.*

Administrator Utilities	Administrator Utilities are a subset of the System Utilities package. They consist of commands used for system administration, such as **Create Configuration File** and **PLog**.
Basic Utilities	Basic Utilities are a subset of the System Utilities package. They consist of commonly used commands, such as **Copy** and **Delete**.
Cluster View Utilities	Cluster View Utilities are a subset of the System Utilities package. They consist of the commands used to start a Cluster View session.
Development Utilities	These commands are used for software development and programming. They are purchased separately from the other Standard Software packages.

DOS Utilities	DOS Utilities are a subset of the System Utilities package. They are commonly used Basic System Utilities named with familiar MS-DOS command names.
Mouse, CD-ROM, Voice, Math, MCR, and DataComm Utilities	All are subsets of the System Utilities package. Their names are descriptive of the commands and system services they contain.
Server Utilities	These commands require interaction with the server.
Sort/Merge Utilities	The Sort/Merge Utilities are a subset of the System Utilities package. They consist of commands used for sorting and merging file contents.
Tape Utilities	These commands provide system services required for tape drive hardware.

Administrator Cluster View

Use the **Administrator Cluster View** command to execute commands on a server from a keyboard and monitor located on a cluster workstation.

The **Administrator Cluster View** command differs from the **Cluster View** command only when the Remote User Manager is running on the processor you want to access. When you start a Cluster View session with **Administrator Cluster View**, no other Cluster View session can be started on that processor. Therefore, an administrator session runs in the primary partition, which is necessary for certain administrative tasks, such as installing system services. See also "Cluster View," in the *CTOS System Administration Guide.*

In addition, the **Administrator Cluster View** command provides access to the Debugger via a pop-up menu.

This command is implemented with the run file *ClusterView.run*, command case 00.

Note: *Access to this command can be granted or denied on a per-user basis. See the* **Install Command Access Service** *command, later in this section, and "Limiting Access to Certain Commands," in the* CTOS System Administration Guide.

Command Form

```
Administrator Cluster View
  [Processor name - XE only]  _____
  [User name]                 _____
  [User file password]        _____
  [Node name]                 _____
  [Old XE run file?]          _____
  [Run file to invoke]        _____
  [Partition size]            _____
```

Parameter Fields

[Processor name - XE only]

Default: Processor to which the workstation is connected

For a workstation server, you can leave this field blank. For an SRP™ server, enter the processor ID for the processor you want to access. Processor IDs consist of a two-character acronym and a two-digit number identifying the position of the processor within the SRP. For example, GP00 is the first General Processor. See "Shared Resource Processor Hardware," in the *CTOS System Administration Guide* for more detailed information about processor IDs.

[User name]

Default: SignOn user name

Enter a user name that is valid for signing onto the server. This means that the user configuration file must exist in *[!Sys]<Sys>* on the server.

[User file password]

Default: None

Enter a valid password for the server.

[Node name]

Default: None

Enter the node name of the server you want to access. If you leave this field blank, you access the server within your own cluster.

[Old XE run file?]

Default: No

Enter **Yes** if the program you want to execute is an old-style run file that does not write to video. Many programs linked pre-12.0 CTOS libraries require this option, but you might not realize in advance that you need to use it. If the program you are executing does not present readable video, exit Cluster View and start the program again, specifying **Yes** in this field. This option does not provide workstation-quality video, but it does provide readable output.

[Run file to invoke]

Default: *[Sys]<Sys>Exec.run*

This field applies only to processors running the Remote User Manager. Enter the name of the run file you want to execute.

[Partition size]

Default: See below

This field applies only to processors running the Remote User Manager. Enter the partition size, in K bytes, to be created for the run file you specified in the preceding field. Default values are 400K bytes for regular Cluster View sessions and all remaining processor memory for administrator sessions.

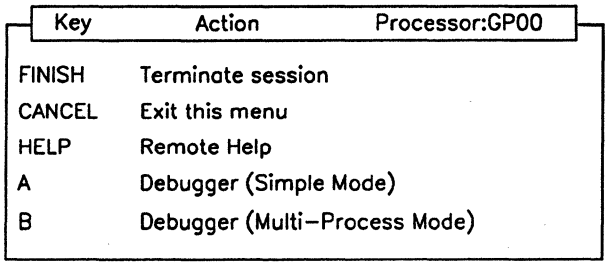
Operation

When you are using Cluster View, your keyboard and monitor are effectively attached to the processor you specified, and the work you perform takes place there. Therefore, *[Sys]<Sys>* refers to the system volume on the server or node you are accessing. The exclamation point (!) is not required because you are actually working on a server. While using Cluster View, you do not have access to your own local disks, communications ports, tape drives, or other peripheral devices.

Menu

While you are working in a Cluster View session, certain functions are performed from the Cluster View menu. The Cluster View menu is pictured in Figure 3-1 and briefly described below.

Figure 3-1. Cluster View Menu



FINISH	Terminates a Cluster View session.
HELP	Displays the Help facility for the application you are using.
CANCEL	Clears the Cluster View menu.
a	Invokes the Debugger in Simple Mode.
b	Invokes the Debugger in Multiprocess Mode.

To invoke the Cluster View menu, press the **HELP** key.

To exit a Cluster View session, press **FINISH** while the Cluster View menu is displayed.

Append

Use the **Append** command to add the contents of one or more files to the end of another file. The first character of each file is appended immediately following the last character of the preceding file. You can append more than one file, either by using wild cards or by entering multiple file names into the command form. You can also use **Append** to print the contents of a file or display it on the screen; see "Device Specifications," in Section 2.

This command is implemented with the Executive internal value !1.

Command Form

```
Append
 File list from  _____________________________
 File to         _____________________________
 [Confirm each?] _____________________________
```

Parameter Fields

File list from

Enter the file specification for the file you want appended to the file named in the *File to* field. You can use a file list, wild cards, or an at-file in this field.

File to

Enter a file specification for the destination file. If the file does not already exist, a new file is created.

[Confirm each?]

Default: No

If you enter **Yes**, you are prompted to press **GO** to confirm or **CANCEL** to deny the append operation for each file, or **FINISH** to terminate the command and return to the Executive command prompt.

Assemble

Use the **Assemble** command to invoke the Assembler, which translates Assembly language source files into to machine-readable code. See the *CTOS Programming Utilities Reference Manual: Assembler* for detailed information about the command form and parameter fields.

Command Form

```
Assemble
 Source files                          _____________________________
 [Errors only?]                        _____________________________
 [Macro expansion (GenOnly)]           _____________________________
 [Object file]                         _____________________________
 [Print file]                          _____________________________
 [Error file]                          _____________________________
 [List on pass 1?]                     _____________________________
 [:f1:]                                _____________________________
 [:f0: ([Sys]<Edf>)]                   _____________________________
```

Bad Sector

Note: *This command is not supported on EISA/ISA IDE drives.*

Use the **Bad Sector** command to move data on a non-SCSI disk from a bad spot to a good area. **Bad Sector** also adds the bad spot you specify to the list of known bad spots for the disk. For **Bad Sector** to relocate data, the disk must contain a contiguous area of disk space large enough to contain the data being moved.

I/O errors (301) signify bad spots on a disk. If you encounter I/O errors, follow these steps to prevent further loss of data:

1. Use the **PLog** command to check the system error log for bad spot information (see the documentation for **PLog**, later in this section).
2. Print the system error log or write down the bad spot information.
3. Back up the disk (see the **Volume Archive** command, later in this section).
4. Execute the **Bad Sector** command on the disk.

This command is implemented with the run file *BadSector.run*.

Command Form

```
Bad Sector
 Device                  _____________________________
 [Volume password]       _____________________________
 [Print file]            _____________________________
 Cylinder                _____________________________
 Head                    _____________________________
 Sector                  _____________________________
```

Parameter Fields

Device

Enter the volume or device name of the disk containing the bad spot.

[Volume password]

Enter the volume password, if any.

[Print file]

Default: None

To record bad spot information in an output file, enter the file name. If the file does not already exist, it is created.

Cylinder

Enter the number of the cylinder containing the bad spot.

Head

Enter the number of the head containing the bad spot.

Sector

Enter the number of the sector containing the bad spot.

Bad Spot Locater

Note: This command is not supported on EISA/ISA workstations.

Use the **Bad Spot Locater** command to display the names of files in which disk bad spots are located. Input to this command includes cylinder, head, and sector numbers for non-SCSI disks, in the correct format for the device. See the **Format Disk** command, later in this section, for information about bad-spot formats.

This command is implemented with the run file *DiskUtilities.run*, command case 05.

Command Form

```
Bad Spot Locater
  [Volume or device name]  _____________________
  [Password]               _____________________
  [Print file]             _____________________
  [LBA]                    _____________________
  [Cylinder]               _____________________
  [Head]                   _____________________
  [Sector]                 _____________________
  [Number of sectors]      _____________________
```

Parameter Fields

[Volume or device name]

Default: Default volume

Enter the volume or device name of the disk on which you want to locate bad spots.

[Password]

Default: Currently active password

Enter the volume password.

[Print file]

Default: Screen only

To keep a record of the file names located, enter a file or printer specification. If the file you specify does not exist, it is created. If it already exists, new output is appended to the existing contents. If you leave this field blank, output is displayed on the screen only.

[LBA]

Default: None

This field is for SCSI disks only. Enter the LBA (logical byte address) of a bad spot that has occurred since the disk was last formatted. The LBAs for such bad spots are identified in the system log file, which can be viewed or printed with the **PLog** command.

Bad spots that were detected during manufacturing and those that were specified during disk formatting have already been mapped to a location elsewhere on the disk. See the **Format Disk** command, later in this section, for more detailed information about SCSI bad spots.

[Cylinder]

Default: None

This field is only for non-SCSI disks and SCSI disks that have not been reformatted since the installation of 3.3 CTOS. Enter the cylinder number of the bad spot.

[Head]

Default: None

This field is only for non-SCSI disks and SCSI disks that have not been reformatted since the installation of 3.3 CTOS. Enter the head number of the bad spot.

[Sector]

Default: None

This field is only for non-SCSI disks and SCSI disks that have not been reformatted since the installation of 3.3 CTOS. Enter the sector number of the bad spot.

[Number of sectors]

Default: 1

Enter the number of contiguous sectors, starting from the number specified in the *[Sector]* field, for which you want bad spots to be located.

Bad Spot Reporter

The **Bad Spot Reporter** command lists the bad block table for a volume. Such a report is useful for reinitializing a disk with a damaged or destroyed bad block table. The contents of a bad block table can be written to a file, which can be printed or stored on a different disk for later use.

This command is implemented with the run file *FormatDisk.run*, command case RR.

Command Form

```
Bad Spot Reporter
 Volume or device name                    _____________________
 [Device password]                        _____________________
 [Volume password]                        _____________________
 [Format Disk configuration file]         _____________________
 [Device template]                        _____________________
 [Print file]                             _____________________
```

Parameter Fields

Volume or device name

Enter the volume or device name of the disk for which you want bad spots reported.

[Device password]

Default: None

To report bad spots on a corrupted volume, enter the device password.

[Volume password]

Default: Currently active password

To report bad spots on a valid volume, enter the volume password.

[Format Disk configuration file]

Default: *[Sys]<Sys>FormatDiskConfig.sys*

To report bad spots on a corrupted volume, enter the name of the **Format Disk** configuration file containing the device template for the disk.

For valid volumes, leave this field blank.

[Device template]

Default: None

To report bad spots on a corrupted volume, enter the name of the device template for the disk. The device template must be contained in the configuration file you specified in the preceding field.

For valid volumes, leave this field blank.

[Print file]

Default: Screen only

To keep a record of the bad spot report, enter a file or printer specification. If the file you specify does not exist, it is created. If it already exists, new output overwrites the existing file.

If you leave this field blank, output is sent to the screen only.

The bad spot report is similar to the defect report provided by the disk manufacturer. It lists the cylinder, head, sector, and size of each bad block on the disk.

Basic ATE

Use the **Basic ATE** (Asynchronous Terminal Emulator) command when you want your workstation to function as an asynchronous terminal. For more information, see the *CTOS Basic ATE User's Guide*.

Command Form

```
Basic ATE
  [Recording filename]          _____
  [Comm channel]                _____
  [Baud rate]                   _____
  [Stop bits]                   _____
  [Parity]                      _____
  [Data bits]                   _____
  [Display mode]                _____
  [Auto linefeed?]              _____
  [Recognize escape code?]      _____
  [Receive timeout]             _____
  [Send Xon/Xoff?]              _____
  [Script filename]             _____
  [Open mode]                   _____
  [Originate?]                  _____
  [Dial string]                 _____
  [Dial time out]               _____
  [Half duplex modem?]          _____
```

Batch Foreground

Use the **Batch Foreground** command to execute programs written in Job Control Language (JCL). For more information, see the *CTOS Batch Manager II Installation, Configuration, and Programming Guide.*

Command Form

```
Batch Foreground
 JCL file name          _____________________________
 [Parameters]           _____________________________
 [Restart JCL file?]    _____________________________
```

Bootstrap

Use the **Bootstrap** command to load an operating system or diagnostics program into memory on a workstation or shared resource processor.

This command is implemented with the run file *Bootstrap.run*.

Command Form

```
Bootstrap
  [File name]                    _____________________________
  [Sys volume or wsNNN]          _____________________________
```

Parameter Fields

[File name]

Default: *[Sys]<Sys>SysImage.sys*

Enter the name of the operating system or diagnostics file you want to load.

[Sys volume or wsNNN]

Default: See below

To boot from a local disk, enter the volume or device name of the disk you want to boot from. To select the local volume appearing in the path setting, enter a period (.). The default is the first device containing a bootable file named *<Sys>SysImage.sys*, as described below:

1. Floppy drives, in numerical order, starting with *f0*
2. QIC tape drives
3. Hard disk drives, in numerical order, starting with *d0*

To boot from *[!Sys]<Sys>WsNNN>SysImage.sys* on the server, enter the three-digit workstation number (*NNN*), as listed in Table 3-1.

Table 3-1. Workstation Type Numbers (Ws*NNN*)

Workstation Number (*NNN*)	Processor	File System
090	Series 3000	Hard disk
092	Series 3000	Diskless
125	B27	Hard disk(s)
126	126	Floppy disk(s) only
127	B27	Diskless
200	B24	Diskless
210	B38* and 386 NGEN®	Hard disk(s)
211	B38* and 386 NGEN	Floppy disk(s) only
212	B38* and 386 NGEN	Diskless
213	Series 5000	Hard disk(s)
219	Series 2000	Diskless
220	B39 and Series 386i™	Hard disk(s)
230	Series 286i™	Hard disk(s)
231	Series 286i	Floppy disk(s) only
240	B28, 286 NGEN, B38*, 386 NGEN*	Hard disk(s)
241	B28, 286 NGEN, B38*, 386 NGEN*	Floppy disk(s) only
242	B28, 286 NGEN, B38*, 386 NGEN*	Diskless
250	B26 and 186 NGEN	Hard disk(s)
251	B26 and 186 NGEN	Floppy disk(s) only
252	B26, CWS, and 186 NGEN	Diskless

*Boot ROM versions higher than 3.2 boot from 210, 211, or 212.; versions lower than 3.2 boot from 240, 241, or 242. The boot ROM version is displayed on the Bootstrap menu; see "Using the Bootstrap Menu," in the *CTOS System Administration Guide*.

Cache Status

Use the **Cache Status** command to obtain information about the file-system cache, an area of memory on protected mode systems, where disk sectors are stored dynamically.

The file-system cache increases the speed at which files are retrieved. See "Using a File System Cache," in the *CTOS System Administration Guide*. See also the related commands, **Enable Caching**, **Disable Caching**, **Lock In Cache**, and **Unlock Cache**, in this section.

This command is implemented with the run file *CacheUtility.run*, command case 00.

Command Form

```
Cache Status
  [File list]
  [Details?]
  [Print file]
```

Parameter Fields

[File list]

Default: See below

Enter the file specifications for which you want to display the cache status. If you leave this field blank, statistics about the cache in general are displayed.

See "Default Display" and "Display of Specified Files," below, for examples of the information displayed by the **Cache Status** command.

[Details?]

Default: No

If you enter **Yes**, the system displays detailed information on the file system cache. See "Details Display," below, for an example.

[Print file]

Default: Screen only

To write screen output to a file or printer, enter a file or printer specification. If the file you specify does not exist, it is created. If it already exists, its contents are overwritten.

If you leave this field blank, output is written to the screen only.

Default Display

The following example shows the **Cache Status** display when you leave the *[File list]* field blank. A brief explanation of each field follows.

```
System Memory Size   :      4194304
Cache Pool Size      :      1057358  ( 25%)
Total Cache Probes   :          126
   Blocks Obtained   :         1009  ( 89%)
              Hits   :          443  ( 43%)
```

System memory size	Size of the host processor's memory, in bytes.
Cache pool size	Size of the file system cache, in bytes. In this example, the cache is on a remote processor.
Total cache probes	Number of times the file system has looked into the cache.
Blocks obtained	Number of cache blocks that have been used for disk access. The percentage shown is a percentage of the total cache probes made.
Hits	Number of times a valid block was obtained from the cache. If the percentage shown is small, the cache is probably too small.

Display of Specified Files

The following example shows a display for specific files. It shows whether the caching attribute is enabled or disabled. The term "system default" reflects the default parameter set in the operating system configuration file.

```
<Sys>CustomDB.run                Disabled
<Sys>Exec.run                    Enabled (system default)
<Sys>QicErase.run                Enabled (system default)
<Sys>TapeRestore.run             Enabled (system default)
<Sys>BackupVolume.run            Enabled (system default)
<Sys>Cluster.run                 Enabled (system default)
<Sys>Assembler.run               Enabled (system default)
```

Note that when a cached file is deleted, it is removed from disk only. Its cached sectors remain in the cache until they are "bumped" by newly cached sectors. Therefore, it is possible that the name of a deleted file will continue to appear in the **Cache Status** display after it has been deleted from disk.

Details Display

The following example shows the information displayed when you specify **Yes** in the *[Details?]* field. A brief explanation of each field follows.

```
System Memory Size       :    4194304
Cache Pool Size          :    1057358 (25%)
     Block Size          :       4096
     Block Count (total) :        256
     Min Working Set Blocks:        0 (0%)

Current Blocks Valid     :        256 (100%)
               Available :         95 (37%)
               Locked-in :        159 (62%)
               Busy      :          2 (2%)

Total Cache Probes       :       1249
   Blocks Obtained       :       1129 (90%)
          Hits           :        562 (49%)
     Blocks Unavailable  :          0 (0%)
```

System memory size	Size of the host processor's memory in bytes.
Cache pool size	Size of the file system cache in bytes.
Block size	Size of the cache blocks in bytes, as specified in the operating system configuration file.example, the cache is on a remote processor.
Block count (total)	Number of blocks allocated in the cache, as specified in the operating system configuration file.
Minimum working set blocks	Minimum number of blocks that cannot contain locked-in data.
Current blocks valid	Number of cache blocks containing valid disk data. After the system has been running, this number eventually goes to 100%. If it does not, the cache is probably too big.

Available	Number of blocks in the cache that can be replaced by disk data.
Locked-in	Number of blocks containing locked-in data.
Busy	Number of blocks currently being used for I/O or DMA. Blocks remain busy for a very brief period of time. This entry is used mainly for diagnostic purposes.
Total cache probes	Number of times the file system has looked into the cache.
Blocks obtained	Number of cache blocks that have been used for disk access. The percentage shown is a percentage of the total cache probes made.
Hits	Number of times a block with the desired disk data was obtained from the cache. If the percentage shown is small, the cache is probably too small.
Blocks unavailable	Number of times the file system looked in the cache but could not obtain a cache block.

Cd

The **Cd** command is identical to the **Path** command. Use it to change the logged node, volume, directory, file prefix, or password.

Cd is provided for users who are already familiar with DOS or UNIX®. See the **Path** command, later in this section, for a detailed description of the command form, parameter fields, and operation of the command.

CFA Configure

Note: *This command is supported in clusters with workstation servers only.*

Use the **CFA Configure** command to create a configuration file for the Cluster File Access (CFA) facility. The Cluster File Access facility provides a data access service, whereby data stored on disks at cluster workstations can be shared throughout the cluster. The configuration file identifies the volume name and access level of disks on individual workstations.

You must execute **CFA Configure** before disks on your workstation are recognized by cluster workstations running the CFA File Filter service. You also use this command to change the access level of a disk on your workstation.

To access a disk via the Cluster File Access facility, you use the unique volume name of the disk (rather than its device name), as shown in the following example:

```
Files
  [File list (*)]  [TechPubs]<Sys>*
```

To access a floppy disk, you specify its device specification followed by the volume name of the system volume. Separate the floppy drive specification and the system volume name with a colon, as shown in the following example:

```
Files
  [File list (*)]  [f0:TechPubs]<Sys>*
```

Note: *Modify access to floppy disks is allowed only if modify access is specified for the system volume. See the description of the* **CFA Configure** *display below.*

See the related commands **CFA Display Volume Information, Install CFA File Filter, Install CFA Workstation Agent,** and **Install CFA Server Service.** See also "Accessing Disks on Cluster Workstations," in the *CTOS System Administration Guide.*

This command is implemented with the run file *CfaWAConfig.run.*

Command Form

```
CFA Configure
 [Config file ([d0]<Sys>CFAConfig.sys)]
```

Parameter Field

[Config file ([d0]<Sys>CFAConfig.sys)]

Enter a file specification for the Cluster File Access configuration file.

Display

An example of the **CFA Configure** display is pictured in Figure 3-2.

Figure 3-2. CFA Configure Display

CFA Cluster Workstation Configurator				
Disk	Volume Name	Access		
D0	CPG–0	Read	Modify	None
D1	CPG–1	Read	Modify	None
F0	F0	Read	Modify	None

541.3–2

Each hard disk on the workstation is identified by its device name in the *Disk* field and by its volume name in the *Volume Name* field. A floppy drive is identified by its device name in both fields. A maximum of 10 devices can be configured.

In the *Access* field, three choices appear:

- *Read* allows files to be opened and read, but not to be modified. A valid password is required.
- *Modify* allows files to be opened and modified. A valid password is required.
- *None* prevents files from being read or modified.

Note: *Modify access to floppy disks is allowed only if modify access is specified for the system volume. In addition, if a password is required to access the system volume, that password is also required to access a floppy disk.*

Operation

Use the **LEFT ARROW** and **RIGHT ARROW** keys to highlight the appropriate access level for each disk.

Press **FINISH** to save the configuration file and exit the command.

CFA Display Volume Information

Note: This command is supported in clusters with workstation servers only.

Use the **CFA Display Volume Information** command to display the names and access levels of the volumes available through the Cluster File Access facility.

For information about setting volume access levels, see the **CFA Configure** command, earlier in this section.

This command is implemented with the run file *CfaVol.run*.

Command Form

```
CFA Display Volume Information
  [Print file]
```

Parameter Field

[Print file]

Default: Screen only

If you want a record of screen output, enter a file or printer specification. If the file you specify does not exist, it is created. If it already exists, its contents are overwritten.

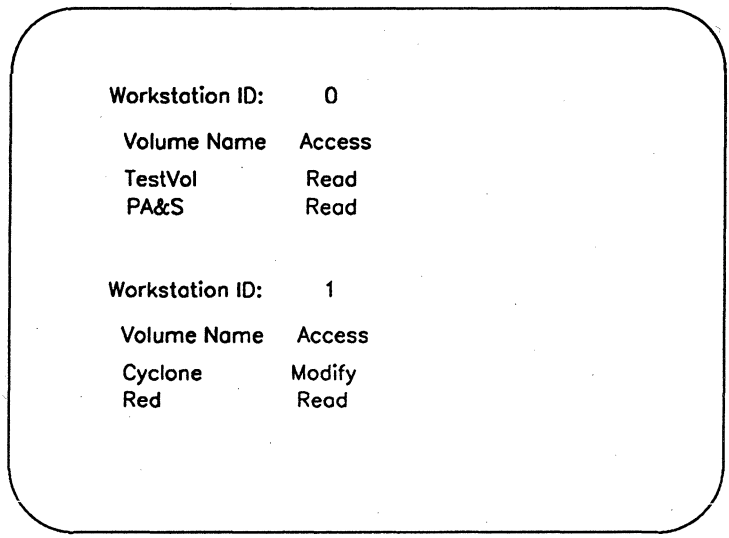
If you leave this field blank, output is sent to the screen only.

Display

Figure 3-3 shows information displayed by the **CFA Display Volume Information** command. Fields are briefly described below.

Workstation ID	An ID number assigned to the cluster workstation
Volume Name	The volume name of a disk on the designated workstation
Access	The access level for the volume

Figure 3-3. CFA Volume Information Display



541.3-3

Change Volume Name

Use the **Change Volume Name** command to change the name of a volume. You can also use this command to add, change, or remove a volume password.

This command is implemented with 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 device name of volume on which you want to change the name.

[Device password]

Default: None

Enter the device password for the disk.

[Old volume password]

Default: Currently active password

Enter the password currently assigned to the volume.

New volume name

Enter the new name you want to assign to the volume. Volume names can contain up to 12 characters.

Notice that this is not an optional field; you must fill it in even if you are changing the volume password only and want to keep the same volume name.

[New volume password]

Default: No change

Enter the new password you want to assign to the volume. Passwords can contain up to 12 characters. If the volume was initialized with password encryption, this new password will be encrypted also. (See the **Format Disk** command, later in this section, for information about password encryption.)

If you leave this field blank, the current volume password remains unchanged.

To remove an existing volume password, enter a pair of single quotation marks ('), as shown in the following example:

```
Change Volume Name
 Device name              d0
 [Device password]
 [Old volume password]    ####
 New volume name          WorkVol
 [New volume password]    ''
```

ChDir

The **ChDir** command is identical to the **Path** command. Use it to change the logged node, volume, directory, file prefix, or password.

ChDir is provided for users who are already familiar with DOS or UNIX. See the **Path** command, later in this section, for a detailed description of the command form and parameter fields.

ChkDsk

The **ChkDsk** command is identical to the **Volume Status** command. Use it to display disk usage information.

ChkDsk is provided for users who are already familiar with DOS or UNIX. See the **Volume Status** command, later in this section, for a detailed description of the command form, parameter fields, and screen displays.

Cluster Status

Use the **Cluster Status** command to display information about cluster line activity. It is frequently used by system administrators to troubleshoot cluster communications problems. It is also used by software engineers to determine the exact cause of cluster communications errors.

Much of the information displayed by **Cluster Status** is intended for interpretation by software engineers, however, it sometimes indicates the cause of problems, such as faulty cabling. See "Error Statistics," later in this section, for a description of errors. See also the *CTOS Cluster and Network Hardware Installation Guide* for more information about cluster communications.

This command is implemented with the run file *ClusterStatus.run*.

Command Form

```
Cluster Status
  [XE line (1)]
```

Parameter Field

[XE line (1)]

If the server is a shared resource processor, enter the number of the cluster line to be displayed first when you execute the command.

Cluster lines are numbered consecutively, starting with 1, for the first Cluster Processor (CP00) or General Processor (GP00) in the system. See "Shared Resource Processor Hardware," the *CTOS System Administration Guide*, for more information.

Once you press **GO**, a function key menu is displayed. You can then cycle through all the lines using **F9** (Prev) and **F10** (Next).

Operation

After you press **GO**, the Errors display appears, as shown in Figure 3-4. From it, you can select function keys to display additional information. To exit **Cluster Status**, press **FINISH**.

Display

The Errors display, as shown in Figure 3-4, is divided into a heading, a status area, and a function key menu. The heading remains in place no matter which function key you select; its fields are described below.

Figure 3-4. Cluster Status Errors Display

Cluster Status 12.2.0 Line Number 01 Mon Jul 27, 1991 1:30 PM
Server Version: vSrvr 1.0.0 Stats Since: Mon Jul 27, 1991 12:00 M
Line Speed: 1.8Mb Line Use Last sec: 0% Line Use Last 10 sec: 0%
WS Total: 24 WS Active: 16 WS Down/Timeout 1 WS Down/Errors 0

ID	User Name	Timeout	CRC	OvRun	Seq	Proto	Addr	Length
00	SERVER TOTALS	14	0	6	0	0	1	0
11	Linnea	0	0	0	0	0	0	0
12	No Name	0	0	6	0	0	0	0
13	June	0	0	0	0	0	0	0
14	WWS	0	0	0	0	0	0	0
15	Tom	1	0	0	0	0	0	0
16	Doug	0	0	0	0	0	0	0
17	Gregg	0	0	0	0	0	0	0
18	Tricia	0	0	0	0	0	0	0
19	Eric	0	0	0	0	0	0	0
1A	Diane	1	0	0	0	0	0	0
1B	Margaret	0	0	0	0	0	0	0
1C	Gloria	0	0	0	0	0	0	0
1D	Ellen	0	0	0	0	0	0	0
1E	Janet	0	0	0	0	0	1	0
1F	Andrew	1	0	0	0	0	0	0
20	Scott	0	0	0	0	0	0	0

541.3-4

Heading

Cluster Status	Displays the version number of the utility.
Line Number	If the server is an SRP, displays the line number for which statistics are currently displayed.
Server Version	Displays the operating system and version number running on the server.
Stats Since	Displays the date and time from which statistics have been collected. This is reset every night midnight or when the system is booted.
Line Speed	Displays the line speed of cluster communications for the currently displayed line.
Line Use Last Sec	Displays the percentage of line usage during the last second.
Line Use Last 10 Sec	Displays the percentage of line usage during the last 10 seconds.
Ws Total	Displays the total number of workstations for which the line is configured.
Ws Active	Displays the number of workstations that are booted and communicating with the server.
Ws Down / Timeout	Displays the number of workstation timeouts that have been reported to the server.
Ws Down / Errors	Displays the number of workstation errors that have been reported to the server.

Function Keys

Additional statistics can be displayed by selecting a function key, as described below.

F1 (Meter)

This function is used by system programmers to tune and debug the cluster. When it is selected, the server begins saving data in a trace buffer allocated by the **Cluster Status** program. When the buffer is full, the following data is displayed:

- Current system time and value of the real-time clock
- Time difference since the last entry
- Event (send, receive, request, response)
- First word of the frame sent or received
- Size of the frame in bytes
- Request code and error returned for frames that contain a request or response

F2 (WsErrs)

This function displays error statistics detected by each workstation. Workstations send error statistics to the server attached to GetDateTime requests. The server saves this information to be displayed by **Cluster Status**. See also "Error Statistics," below.

F3 (Errors)

This function displays error statistics detected by the server. Errors are shown for each workstation, as well as a total for all workstations. When errors occur, it is important to know which workstations are having problems. See also "Error Statistics," below.

F4 (Data)

This function displays the following information about each workstation:

- Number of requests processed
- Number of sector pages read
- Number of sector pages written

These statistics show which workstations are using cluster lines heavily. On shared resource processors, such statistics can help the system administrator balance the work load among processors.

F5 (Blocks)

This function displays the following statistics to show how X blocks, Y blocks, and Z blocks are being used. Note that Y blocks and Z blocks relate only to SRP servers.

- Number of each block type allocated
- Number of each block type currently free

The following statistics appear for X blocks only:

- Sizes of X blocks as specified when the operating system was built (workstations only) or in *[Sys]<Sys>SrpConfig.sys*.
- Number of usable blocks for each block type
- Number of X-block waits (the number of times the system could have used more X blocks)

These statistics are of assistance in tuning the system. If the number of X block waits is greater than 0, system performance is degraded. See also the *CTOS System Administration Guide*.

F6 (Frames)

This function displays statistics about the following types of frames that have been sent and received by the server:

- Receiver Ready
- Receiver Not Ready
- IFrame information
- Set Normal Response Mode (SNRM)
- Number of the last SNRM sent

An SNRM announces a station address that is not being used. A workstation that is waiting for a station address, so that it can come on-line, may respond to a SNRM.

Such information is used by software engineers to determine the cause of cluster line problems.

F7 (Time)

This function displays the following timing statistics:

- How many times each workstation has been polled
- Number of milliseconds since the last poll
- Maximum time between polls for each workstation
- Average time interval between polls

These statistics can be used to monitor response time of the server.

F8 (RqTime)

This function displays timing statistics for request processing. For each workstation, the following information is displayed:

- Cluster overhead
- Time to process a simple request
- Time to process a blocked request

A blocked request is a request that must be broken up because the data is too large to fit into one X block. This statistic is calculated by the Workstation Agent and is attached to each GetDateTime request sent to the server.

The "cluster overhead" statistic is derived from the time it takes to queue a GetDateTime request, send it to the server, and receive the response. It is a good measure of cluster overhead.

F9 (Prev) and **F10** (Next) — These keys cycle through the cluster lines on an SRP server.

Error Statistics

Different types of errors, as displayed by **Cluster Status**, are described below. See also the *CTOS System Administration Guide* and the *CTOS Cluster and Network Hardware Installation Guide*.

Timeout — These errors occur when a frame is sent to a receiving station, but no reply is received during the specified interval. In cabled clusters, timeouts are usually caused by cabling or hardware problems, for example, if a cluster cable is unplugged. With Tele-Cluster, however, timeouts are normal.

CRC errors — These errors occur when the cyclic redundancy check (CRC) at the end of the frame does not match the CRC calculated by the chip when the frame is received. CRC errors can result from hardware problems, improperly terminated lines, or more than one master in a cluster. Expect to see these when adding or removing workstations from a daisy-chained cluster, or when workstations are in the process of booting.

Overrun errors	These errors occur when the sending station has a transmit overrun, which causes the receiving station to set a termination bit. Overrun errors are most commonly caused by a failure with communications hardware.
Sequence errors	These errors occur when an information frame (IFrame) is lost or out of sequence. An IFrame can be lost if a CRC error is detected by the receiving station. See also "CRC Errors," above.
Protocol errors	These errors occur when a station receives a frame it is not expecting. This can happen if there is more than one server on the same cluster line.
Address errors	These errors occur when the server polls a workstation and receives a response from a different workstation. This situation occurs most often when there is a hardware problem, for example, if the cluster is not properly terminated. They can also occur when workstations are booting.
Length errors	These errors occur when the number of bytes sent does not match the number of bytes received. These errors can result from hardware or cabling problems. They can also be a by-product of CRC, sequence, protocol, or address errors.

Cluster View

Use the **Cluster View** command to execute commands on a server from a keyboard and monitor located on a cluster workstation.

The command form, parameter fields, and operation of the **Cluster View** command are identical to the **Administrator Cluster View** command. See its documentation, earlier in this section. See also "Cluster View," in the *CTOS System Administration Guide*.

This command is implemented with the run file *ClusterView.run*, command case 01.

Note: *Access to this command can be granted or denied on a per-user basis. See the* **Install Command Access Service** *command, earlier in this section, and "Limiting Access to Certain Commands," in the* CTOS System Administration Guide.

Command File Editor

Use the **Command File Editor** command to add, modify, or remove Executive commands. You can also use it as a tool for creating submit files and Job Control Language (JCL) files.

For information about Executive command files, see "Command Files," in Section 2. For information about submit files, see the **Submit** command, later in this section. For information about JCL files, see the *CTOS Batch Manager II Installation, Configuration, and Programming Guide*.

This command is implemented with the run file *CmdFileEditor.run*, command case 00.

Command Form

```
Command File Editor
  [Command file]  _____________________________________
```

Parameter Field

[Command file]

Default: The command file you are currently using

Enter the file specification of the command file you want to edit.

Operation

After you fill in the command form and press GO, the **Command File Editor** verifies that the file you specified is a command file. If it is not, you are prompted to press GO to exit the command.

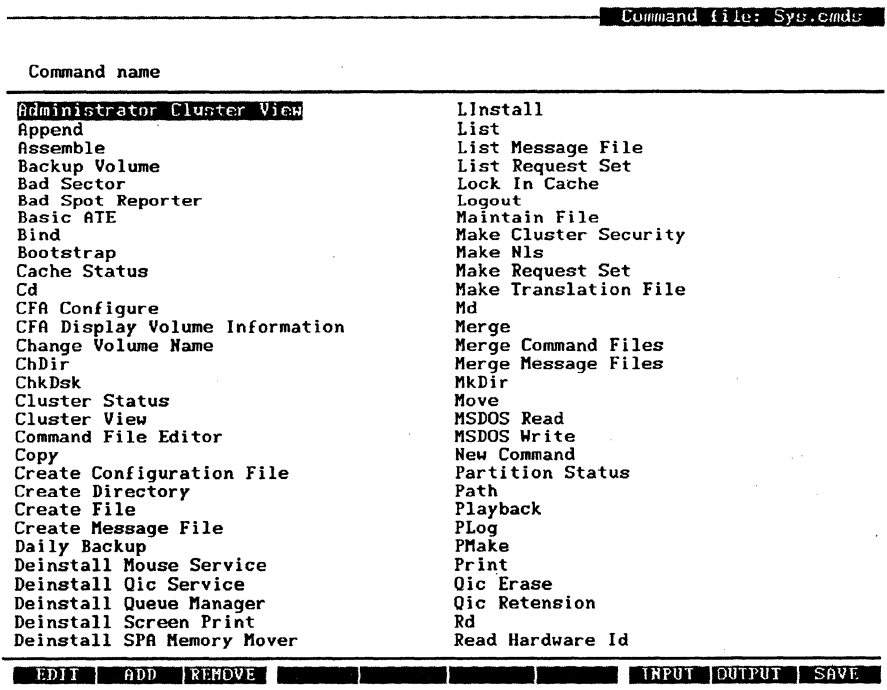
If the file does not exist, you are informed and asked if you want to create a new command file. Press GO to confirm or CANCEL to exit the **Command File Editor.**

To exit the **Command File Editor**, press FINISH, then GO.

Display

A sample **Command File Editor** display is pictured in Figure 3-5. If the command file contains more command names than will fit on the screen, press **NEXT PAGE** or **SCROLL UP** to display more command names.

Figure 3-5. Command File Editor Display



541.3-5

Function Keys

A function key menu is displayed at the bottom of the screen. Use these keys as follows:

F1 (Edit)	To modify or view the command form, run file, command case, and Help description for the highlighted command
F2 (Add)	To add a new command
F3 (Remove)	To remove the highlighted command
F8 (Input)	To copy a command from another command file
F9 (Output)	To create or append a command to a submit file, a JCL file, or an ASCII text file
F10 (Save)	To save changes you make to the command file

Adding Commands

To create a new command, follow these steps:

1. Press F2 (Add); the following form appears at the top of the screen:

   ```
   Command name    _____________________________________
   Copy from command _____________________________________
   ```

2. Type a name for the new command in the *Command name* field.
3. Optionally, press RETURN and fill in the *Copy from command* field with an existing command name.

   This copies the command form, run file, command case, and Help description from an existing command, which saves time if the new command is similar to another.

4. Press GO. Blank fields for a command form, run file, command case, and Help description are displayed.

5. Fill in a command form, run file, command case, and Help description.

   To do so, position the highlight on the field you want to fill in. Use **RETURN**, **DOWN ARROW**, or **UP ARROW** to move the highlight one line or field at a time. Press **CODE-RETURN** to move the highlight to the Run file field near the bottom on the screen.

   BACKSPACE, **DELETE**, and **CODE-DELETE** work in the usual manner (see "Keyboard," in Section 2).

6. Press **GO** to save the new command. You are prompted to enter **Y** to confirm or **N** to discard the changes.

You can terminate the Add function at any time by pressing **CANCEL**.

Removing Commands

To remove a command, follow these steps:

1. Select a command by positioning the highlight on the command name.
2. Press **F3** (Remove).
3. Press **GO** to confirm that you want to remove the command, or press **CANCEL** to terminate the Remove function.

Modifying Commands

Follow these steps to change a command name, a parameter field, the run file, command case, or Help description:

1. Select a command by positioning the highlight on the command name.
2. Press **F1** (Edit).
3. Select a field by positioning the highlight on the field you want to edit.
4. Edit the field.

   BACKSPACE, **DELETE**, and **CODE-DELETE** work in the usual manner (see "Keyboard," in Section 2).

5. Save your changes.

   When you have finished editing, save your changes by pressing **GO**. You are prompted to enter **Y** to confirm or **N** to discard the changes.

Note that if the last character of a field name is an asterisk (*), expansion of wild card characters is suppressed when a wild card is used as a parameter value in that field. The asterisk in the field name does not appear when the command form is displayed by the Executive. Likewise, the circumflex (^) can be used in password fields to camouflage password characters.

The **Command File Editor** verifies that all required fields are filled in. If either the *Command name* or *Run file* field is left blank, the workstation beeps and the highlight returns to the empty field. You must fill in the field before you can save changes.

You can terminate the Edit function at any time by pressing **CANCEL**.

Copying Commands

To copy a command that exists in a different command file, follow these steps:

1. Press **F8** (Input) to display the following form:

   ```
   Command name  _____________________________________
   Command file  _____________________________________
   ```

2. Type the name of the command you want to copy in the *Command name* field.
3. Press **RETURN**.
4. Type the name of the command file that contains the command you want to copy in the *Command file* field.
5. Press **GO**.
6. Edit the command information, if necessary.
7. Press **F7** (Return) or **GO** to save the command.

You can terminate the Input function at any time by pressing **CANCEL**.

Each time you use the Input function, the *Command file* field is filled in with the name of the command file that you previously used. To change the command file name, clear the field with **CODE-DELETE**.

Saving Changes

The **F10** (Save) function key saves the changes you have made to a command file. Use **F10** (Save) to save changes at any time, without exiting from the **Command File Editor**. Also, your changes are automatically saved when you press **FINISH**, then **GO**, to exit the **Command File Editor**.

If the command file you are editing is in use, it cannot be changed and an error message is displayed. That can happen if the command file is in use by another Context Manager partition or if it is located on the server and is being used by another workstation. In either case, the open command file must be closed before you can save any changes you have made to it.

Output Files

With the **Command File Editor**, you can create files in the following formats:

- Submit files to execute commands
- Submit files to create new commands
- Unformatted ASCII text files
- JCL files

To create an output file, follow these steps:

1. Position the highlight on the first command you want placed in the output file.
2. Press **F9** (Output). The following form is displayed at the top of the screen:

```
Command name:  CreateDirectory_____                    Params? Y
Output file:   _____________________                   Append? Y
Output file format:S S=Submit, N=NewCommand, U=Unformatted, J=JCL
```

   Notice that the **OVERTYPE** key lights up. You must be in overtype mode to complete this form.

3. Fill in each field of the form (see "Output Fields," below).

   Use **RETURN** or **UP ARROW** to position the cursor.

4. Press GO.

   If you are creating a submit or JCL file, and entered **Y** in the *Params?* field, the parameter fields for the command are displayed.

5. Fill in parameter values; they will be recorded in the output file.

6. Press GO.

Repeat steps 1 through 6 to include additional commands in the output file. For subsequent commands, you do not need to fill in the form shown in step 2.

Output Fields

Command name

This field is automatically filled in with the command name that was highlighted when you selected F9 (Output). If you inadvertently selected the wrong command, press CANCEL to select another command, or press CODE-DELETE to clear the field, then type the command name.

Params?

If you are creating a submit or JCL file, enter **Y** to include parameter values in your output file. If you enter **N**, parameter fields are left blank.

Output file

Enter a file specification for the output file. If you are outputting an unformatted file, you can specify a printer in this field.

Append?

Enter **Y** to append the command output to an existing file. Enter **N** to overwrite an existing file.

Output file format

Enter one of the following:

s for a submit file

n for a submit file that creates a new command

u for an unformatted ASCII file

j for a JCL file

(Sample output file formats are shown below.)

Example of a Submit Output File

The following example shows a submit output file containing parameter values. It can be executed as is, without requiring manual editing. See the **Submit** command, later in this section, for detailed information about submit files.

```
%'
Create Directory
%'New directory name(s)'NewDir
%'[Default protection level (15)]'5
%'[Maximum number of files (75)]'
%'[Password for new directory]'jes
%'[Volume password]^'12354
•%'
Path
%'[Volume]*'
%'[Directory]*' NewDir
%'[Default file prefix]*'
%'[Password]^'jes
%'[Node]*'
•
```

Example of a New-Command Output File

The following example shows an output file that creates a new command when executed with the **Submit** command.

```
%'
Copy
[kbd]
[Scr]<$>args.Tmp
Yes• 'Volume or device name' '[Volume or device
password]^' '[Bad spot file]' '[Print file]'
(FINISH symbol appears here)
Copy
[kbd]
[Scr]<$>desc.Tmp
Yes•
'Report bad spots on a volume'
(FINISH symbol appears here)
New Command
'Bad Spot Reporter'
[Sys]<Sys>BadSpotReport.run
@[Scr]<$>args.Tmp
@[Scr]<$>desc.Tmp
Yes
'ø0ø0•
```

Example of an Unformatted Output File

The following example shows an output file containing commands names and command forms only. It is not an executable file.

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

Delete
  File list
  [Confirm each?]
```

Example of a JCL Output File

The following example shows a JCL output file.

```
; $Job JobName, UserName, [Password], [Sysout file]
$Run ([Sys]<Sys>Mouse.run, 'Install Mouse Service', '00')

$Run ([Sys]<Sys>FontService.run, 'Install Font Service',&
     '00'),[Sys]<Gps>Font.dbs
```

To make it executable with the Batch facility, make the following changes with a text editor:

1. Uncomment the $Job statement by removing the leading semicolon.
2. Edit the parameters for the $Job statement, so that they are appropriate for the job you want to execute (see the *CTOS Batch Manager II Installation, Configuration, and Programming Guide*).
3. Add an $End statement to the end of the JCL file.

Configure Sequential Access Device

Note: This command is not supported on EISA/ISA workstations.

Use the **Configure Sequential Access Device** command to create or modify a configuration file for a quarter-inch cartridge (QIC), digital data storage (DDS), or half-inch tape drive.

Tape-drive configuration files are optional. By default, the Sequential Access Service configures tape drives to provide good performance in a variety of settings. For optimal performance, however, you may need to modify the configuration file for your tape drive.

This command is implemented with the run file *SeqAccessUtility.run*, command case CF.

Command Form

```
Configure Sequential Access Device
 Configuration file name _____________________________
```

Parameter Fields

Configuration file name

Enter the file specification for the configuration file you want to create or modify. To be recognized by the Sequential Access Service, tape configuration file names must be constructed in the following format:

*[Sys]<Sys>*DeviceName*Config.sys*

where *DeviceName* is the device name specified to the Sequential Access Service (see the **Install Sequential Access Service** command, later in this section). For example, the configuration file name for a tape drive named QIC is *[Sys]<Sys>QICConfig.sys*.

Operation

After you press GO, a list of configurable parameters is displayed. If you are creating a new configuration file, default values are filled into the fields, as shown below. If you are modifying an existing file, current values appear in the fields.

```
Sequential Access Device Parameters
  [Minimum (or fixed) record size (512)]                          512
  [Maximum record size (variable records)]                        512
  [Block size (fixed records, defaults to record size)]           512
  [Recording density code]                                        0
  [Transport speed code]                                          0
  [Service buffer size, in bytes]                                 0
  [Total service buffers]                                         0
  [Unbuffered mode?]                                              No
  [Erase to EOM after close?]                                     No
  [Rewind after close?]                                           No
```

To exit the command and save changes, press GO. To exit the command without saving changes, press CANCEL. Configurable parameters are described below.

Configurable Parameters

[Minimum (or fixed) record size (512)]

Default: 512 bytes

Specify the record size, in bytes, for fixed-size records; or enter the minimum size, in bytes, for variable-length records. If applicable, specific record size and type requirements should be documented for the application that is writing data to the tape. For example, the Standard Software archive utilities (**Restore Archive**, **Selective Archive**, and **Volume Archive**) require 512-byte fixed records.

[Maximum record size (for variable records)]

Default: 512 bytes

Enter the same value you specified as the minimum record size for fixed-size records; or, for variable-length records, enter the maximum size, in bytes.

[Block size (fixed records, defaults to record size)]

Default: Same as record size

For QIC and DDS tape drives, set this value to **512**. Archive tapes created with varying block sizes will not be readable from drive to drive.

For half-inch tape drives, enter an even multiple of the record size (for example, 4096) for fixed-length records. The maximum value is 65535. For variable-length records, enter **0**.

[Recording density code]

Default: 0

This parameter pertains to SCSI half-inch tape drives only. Enter one of the following codes:

1	800 bpi NRZI
2	1600 bpi PE
3	6250 bpi GCR
6	3200 bpi PE

See the documentation for your tape drive to find out which recording densities are supported.

[Transport speed code]

Default: 0

This parameter pertains to half-inch tape drives only. For non-SCSI half-inch tape drives, **0** specifies the lower supported transport speed, **1** the higher. For SCSI half-inch tape drives, specify **0**.

[Unbuffered mode?]

Default: No

Enter **Yes** to use the tape drive in unbuffered mode. Enter **No** to use the tape drive in buffered mode.

[Service buffer size, in bytes]

Default: 0

Enter the number of bytes of memory to be allocated for each Sequential Access Service buffer. This number must be a multiple of the block size for fixed records, or at least as large as the maximum record size for variable records.

The default of **0** indicates that an optimum buffer size for the specific device should be used. Note that if this parameter is set to **0**, the following field (*[Total service buffers]*) must also be set to **0**.

[Total service buffers]

Default: 0

Enter the number of buffers to be used by the Sequential Access Service.

The default of **0** indicates that an optimum number of buffers for the specific device should be used. Note that if this parameter is set to **0**, the preceding field (*[Service buffer size, in bytes]*) must also be set to **0**.

Caution

Do not use the following parameter with QIC tape drives; doing so erases the entire QIC tape. Use the following parameter with half-inch and DDS tape drives only.

[Erase to EOM after close?]

Default: No

Enter **Yes** if you want to erase the tape to the end of the medium (EOM) after a dataset is written. This removes extraneous data from a previously used tape. Note, however, that this option adds time to tape procedures; in the case of DDS drives, that could be several hours.

Enter **No** if you do not want to erase the tape.

[Rewind after close?]

Default: No

Enter **Yes** if you want to rewind the tape after a dataset is written.

Enter **No** if you do not want to rewind the tape.

Convert Nls.sys

Use the **Convert Nls.sys** command to convert the keyboard data in an *Nls.sys*-type file to a new-style keyboard data block, an object module, or an editable text file. This is necessary if you have previously customized or internationalized an *Nls.sys* file and want to reuse its contents in the system keyboard table.

Nls.sys files are used with model K1 keyboards. See the **Convert Sys.keys** command, later in this section, for information about models K2 to K5 keyboards.

See the **Create Keyboard Data Block** command for information about incorporating customized keyboard data into *NlsKbd.sys*. See "Customizing the System Keyboard Data Blocks," in the *CTOS Operating System Concepts Manual*, for information about customizing keyboard tables.

This command is implemented with the run file *KCT.run*, command case 22.

Command Form

```
Convert Nls.sys
 Nls.sys file name
```

Parameter Fields

Nls.sys file name

Enter the file specification of the *Nls.sys*-type file you want to convert.

Operation

After you press **GO**, an interactive screen is displayed. Press one of the following function keys to select one or more types of command output:

F1 (None)	Exits the command.
F2 (.obj Only)	Outputs an object module named *KnKbd.obj*, where *Kn* is the keyboard model number (for example, K1).

F3 (.tbl Only)	Outputs a binary keyboard table to a file named *KnKbd.tbl*.
F4 (Update Only)	Outputs an editable text file named *KnKbd.kbd_data*.
F6 (.obj + .tbl)	Outputs both an object module and a binary keyboard table.
F7 (.tbl + Update)	Outputs both a keyboard table and an editable text file.
F8 (.obj + Update)	Outputs both an object module and an editable text file.
F10 (All Files)	Outputs an object module, a binary keyboard table, and an editable text file.

If you select text file output only (**F4**), the text file is created and the command exits. If you select any other type of output, the following parameter form is displayed:

```
Output file name                    _____________________________
Type keyboard modified              _____________________________
IBus ID                             _____________________________
Include untranslate table?          _____________________________
PUBLIC in object module             _____________________________
```

To edit the form, press **F3** (Edit Above). Other function keys are **F1** (Exit NoSave), which exits the command without saving changes; **F5** (Only Trans), which creates a file with translation information only; **F7** (Only Emul), which creates a file with emulation information only; and **F9** (All Data), which creates a file using all data in the *.kbd_data* file.

Parameter fields are described below. Note that the last parameter appears only when you select object module output.

Output file name

This field is initially filled in with the default file name of the output type you selected (see the function key descriptions, above). If you want to change the default file name, enter the file name you want to assign to the file.

Type keyboard modified

This field is initially filled in with the current keyboard model (such as K1). If this is not correct, enter the appropriate keyboard type.

IBus ID

Enter the IBus identification number for the keyboard. This field is initially filled in with the IBus value for the current keyboard.

Include untranslate table?

Enter **Yes** to create a "untranslation table," for use when keyboard data is passed between two partitions that read different keyboard tables. (An untranslation table is large and adds considerably to the size of the keyboard table file. It does not, however, affect performance of the keyboard table, so file-size considerations are the only reason not to include it.)

PUBLIC in object module

This field appears only if you selected object module output. It is initially filled in with the string KeyBoardData. Enter the string you want to use as the PUBLIC name of the first byte of the table in the outputted object module.

Convert Public Case

Use the **Convert Public Case** command to convert public, communal, and external symbols in an object module or library to a specified capitalization scheme prior to case-sensitive linking. In addition, this command marks libraries as case sensitive.

This command is implemented with the run file *PublicSymbolUtility.run* command case CS.

Command Form

```
Convert Public Case
 Input file                                  _____
 [Output file]                               _____
 [Input case sensitive symbol file]          _____
 [Output symbol file]                        _____
 [Print file]                                _____
```

Parameter Fields

Input file

Enter the file specification of the object module or library you want to convert.

[Output file]

Default: See below

Enter a file specification for the converted file. If you leave this field blank, the output file is written to the file specification of the input file, and the suffix *-old* is appended to the file name of the original input file.

[Input case sensitive symbol file]

Default: None

Enter the file specification of the symbol file containing correctly capitalized symbols. In this symbol file, text preceded by a semicolon (;) is treated as a comment and is ignored.

If you leave this field blank, symbols encountered in the input file are written to the print file, which can be later edited to create a case sensitive symbol file. See the *[Print file]* parameter, below, for a description of what the print file contains.

[Output symbol file]

Default: None

Enter a file specification for the output symbol file. Such a file contains a list of all public, communal, and external symbols that are encountered in the input object module or library.

If a symbol is defined as a public symbol, the comment ";Code" or ";Data" is appended to the public symbol from the input library or object module to indicate whether it is located in a code segment or a data segment. If a symbol is defined as external or communal, no comment is appended.

[Print file]

Default: Screen only

Enter a file specification for the print file. The print file contains the name of the object module or library, followed by the names of symbols that were not listed in the case-sensitive input file.

Convert Sys.keys

Use the **Convert Sys.keys** command to convert a *Sys.keys*-type file to a new-style keyboard data block, an object module, or an editable text file. This is necessary if you have previously customized or internationalized a *Sys.keys* file and want to reuse its contents in the system keyboard table.

Sys.keys files are used with models K2 to K5 keyboards. See the **Convert Nls.sys** command, earlier in this section, for information about the model K1 keyboard.

See the **Create Keyboard Data Block** command for information about incorporating your customized keyboard data into *NlsKbd.sys*. See "Customizing the System Keyboard Data Blocks," in the *CTOS Operating System Concepts Manual*, for information about customizing keyboard tables.

This command is implemented with the run file *KCT.run*, command case 11.

Command Form

```
Convert Sys.keys
  Sys.keys file name _____________________________
```

Parameter Fields

Sys.keys file name

Enter the file specification of the *Sys.keys*-type file you want to convert.

Operation

Operation of this command is identical to the **Convert Nls.sys** command. See the documentation for that command, earlier in this section.

Copy

Use the **Copy** command to copy the contents of one file to another file. You can copy more than one file, either by using wild cards or by entering multiple file names in the command form. You can also use **Copy** to print the contents of a file or display it on the screen; see "Device Specifications," in Section 2.

This command is implemented with the Executive internal value !2.

Command Form

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

Parameter Fields

File from

Enter the file specification for the file you want to copy. You can specify more than one file by using wild cards or by leaving a space between file specifications.

File to

Enter a file specification for the destination file.

If you use wild cards, they must match, in number and pattern, those you entered in the *File from* field.

If you entered more than one file specification in the *File from* field, you must enter the same number of file specifications here. The first file named in the *File from* field is copied to the first file specification in this field, and so on.

[Overwrite ok?]

Default: Ask for confirmation

If you enter **Yes** and the destination file already exists, its contents are replaced with those of the file you are copying. If you enter **No** and the destination file already exists, the file is not copied. If you leave the field blank, you are prompted to press **GO** to confirm overwriting an existing file.

[Confirm each?]

Default: No

If you enter **Yes**, you are prompted to press **GO** to confirm or **CANCEL** to deny the copy operation for each file, or **FINISH** to terminate the command.

If you enter **No** or leave this field blank, files are copied without user interaction.

Create Backup Floppy

Note: This command is not supported on EISA/ISA workstations.

Use the **Create Backup Floppy** command to create a set of diskettes from which you can bootstrap a workstation and then perform the following tasks:

- Sign on
- Back up disks to tape or floppies
- Restore a backup
- Initialize a disk
- Change the name of a volume

This command requires a special Executive message file named *[Sys]<Sys>CBF>ExecMsg.bin*, which contains messages 32907 through 32957 from *[Sys]<Sys>ExecMsg.txt*. If that file does not exist, use the **Create Message File** command to create it. See the documentation for that command, earlier in this section; see also the *CTOS System Administration Guide*, for a detailed procedure.

This command is implemented with the run file *SubCmd.run* and the submit file *CreateBackupFloppy.sub*.

Command Form

```
Create Backup Floppy
  [Include tape utilities?]  _____
  [High capacity floppies?]  _____
```

Parameter Fields

[Include tape utilities?]

Default: No

Specify **Yes** if you want the Sequential Access Service utilities to be included on the diskette set. If you do not want to include the tape utilities, leave this field blank.

[High capacity floppies?]

Default: No

The following possibilities exist for this field:

- If the workstation is equipped with a 3.5-inch drive, specify **Yes.** (If you leave this field blank, the command will fail.)
- If the workstation is equipped with a regular-capacity 5.25-inch drive, leave this field blank. (If you specify anything in this field, the command will fail.)
- If the workstation is equipped with a high-capacity 5.25-inch drive, and you want to create a high-capacity diskette set, specify **Yes.**
- If the workstation is equipped with a high-capacity 5.25-inch drive and you do not want a high-capacity diskette set, leave this field blank.

Note: *Because* **Create Backup Floppy** *is executed from a submit file, any response in a field activates that parameter. Therefore, to respond* **No** *to either field, leave it blank.*

Operation

After some introductory messages are displayed, you are prompted to insert the first diskette. Instructions will appear on the screen; continue responding to messages and changing diskettes as prompted.

As the diskettes are created, label them sequentially, as follows:

Regular Capacity Set

Boot Diskette
Continuation Diskette 1: Optional Tape Services *(You will need to use this diskette even if you did not include tape utilities.)*
Continuation Diskette 2: Volume Archive, Change Volume Name
Continuation Diskette 3: Format Disk
Continuation Diskette 4: Restore Archive

High Capacity Set

Character or Bitmap Boot Diskette
VGA Boot Diskette
Continuation Diskette

Using the Bootable Diskette Set

To bootstrap a workstation from the diskette set, follow these steps:

1. Turn on the workstation.
2. Insert the Boot Diskette into the floppy disk drive.
3. Press the Reset button.

   When the workstation has successfully booted, the following message appears:

   ```
   Please insert Continuation 1 Diskette

         Press any key to continue
   ```

4. Remove the Boot Diskette.
5. Insert the Continuation 1 Diskette.
6. Press GO.
7. Insert the appropriate continuation diskette, as described on the screen.
8. Press GO.

With the regular-capacity diskette set, you will need to change diskettes between backup, format, and restore procedures.

In addition to commands already described, the following commands are available on the continuation diskettes:

Batch Foreground
Copy
Create Directory
Create File
Delete
List
Logout
Path
Rename
Run
Type

Create Boot Tape

Use the **Create Boot Tape** command to create a bootable tape for shared resource processors. A bootable tape contains the files required to initialize a system volume in memory on a shared resource processor running the CTOS/XE operating system, version 3.0 or later.

Before using this command, you create a Table of Contents file for the tape. That file specifies the operating system to boot on each processor, as well as files to restore to the memory disk. The number of files you restore, hence the functionality of the memory disk, is generally determined by the amount of memory available on the shared resource processor.

In many cases, the bootable tape distributed with Standard Software is appropriate for software installation and troubleshooting. You might create your own tape to boot from a customized operating system or to add utilities to the memory disk.

This command is implemented with the run file *CreateBootTape.run*.

Command Form

```
Create Boot Tape
  [QIC device ([QIC])                    _____
  [TOC file (SrpTable.sys)]              _____
  [MemDisk device ([M0])]                _____
  [Format MemDisk overwrite ok?]         _____
```

Parameter Fields

[QIC device ([QIC])]

Default: *[QIC]*

Enter the device name for the tape drive you will use to create the boot tape.

[TOC file (SrpTable.sys)]

Default: *[Sys]<Sys>SrpTable.sys*

Enter the file specification for the Table of Contents (TOC) file you want placed on the bootable tape. **Create Boot Tape** copies this file to tape mark 1. See "Table of Contents File," below, for information about creating a TOC file.

[MemDisk device ([M0])]

Default: *[m0]*

Enter the device name of the memory disk you want to use for the memory disk image. The device name you specify here must be defined with the *:MassStorage:* parameter in the operating system configuration file. See the *CTOS System Administration Guide.*

[Format MemDisk overwrite ok?]

Default: Yes

This field pertains to the memory disk you specified in the *[MemDisk device]* field. Enter **Yes** if you want the **Format Disk** command to reinitialize the memory disk, even if it is currently a valid volume.

If you enter **No** or leave this field blank, you are prompted to confirm reinitializing a valid memory-disk volume.

Operation

When you execute the **Create Boot Tape** command, the following steps take place without user intervention.

1. The memory disk is initialized, and the specified list of files is copied to it.
2. The loader, which boots the proper operating system for the master processor, is copied to tape mark 0.
3. The TOC file is copied to tape mark 1.
4. The operating systems, memory disk images, and archive files, as specified in the TOC file, are copied to the tape.

Table of Contents File

The Table of Contents file is read by the **Create Boot Tape** command to copy operating systems, memory disk images, and archive files to the tape. To create a customized boot tape, you must create a Table of Contents file.

The Table of Contents file is formatted as follows:

:FileType: Token
:FileName: FileSpec

where

Token	Is a specific token value, as listed below.
FileSpec	Is a file specification.

File-Type Tokens

The token values *tocSrpBt* and *tocSrpTable* must be first and second, respectively, in the TOC file. Other token values may appear in any order. With the exception of *tocTapeArchive*, each token value may appear only once per TOC file.

tocSrpBt

This file type specifies the loader, which boots the correct operating system on the master processor. It must be the first entry in the TOC file.

tocSrpTable

This specifies the Table of Contents file. It must be the second file type in the TOC file.

tocSrpGp

This specifies the operating system to boot on General Processors (including GP+SI and GP+CI).

tocSrpFp

This specifies the operating system to boot on File Processors.

tocSrpDp

This specifies the operating system to boot on Data Processors.

tocSrpCp

This specifies the operating system to boot on Cluster Processors.

tocSrpSp

This specifies the operating system to boot on Storage Processors.

tocSrpTp

This specifies the operating system to boot on Terminal Processors.

tocMemDisk1

This specifies the list of files to be stored in a 1-megabyte memory disk.

tocMemDisk3

This specifies the list of files to be stored in 3-megabyte memory disk.

tocTapeArchive

This specifies the list of files to be restored to the hard disk being formatted for the system volume.

Example

An example TOC file is shown below.

```
:FileType: tocSrpBt
:FileName: <Gen>SrpBt.run

:FileType: tocSrpTable
:FileName: <OsInstall>SrpTable.sys

:FileType: tocSrpGp
:FileName: <Sys>pSrpGp.img

:FileType: tocSrpFp
:FileName: <Sys>rSrpFp.run

:FileType: tocSrpDp
:FileName: <Sys>rSrpDp.run

:FileType: tocSrpCp
:FileName: <Sys>rSrpCp.run

:FileType: tocMemDisk1
:FileName: <Mem>MemDisk1.fls

:FileType: tocMemDisk3
:FileName:<Mem>MemDisk3.fls

:FileType: tocTapeArchive
:FileName:<Sys>Archive1.fls

:FileType: tocTapeArchive
:FileName: <Custom>Archive2.fls
```

Create Configuration File

Use the **Create Configuration File** command to create configuration files for the following devices:

- Serial printers
- Parallel printers
- Communication devices

For information about creating configuration files for tape drives, see the **Configure Sequential Access Device** command, earlier in this section.

This command is implemented with the run file *CreateConfig.run*.

Note: *If you are using the Generic Print System™ (GPS), you do not need to create configuration files for your printers.*

Command Form

```
Create Configuration File
 Configuration file name  _____________________
 Device type              _____________________
```

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

Enter the device code, as listed below, for the type of configuration file you want to create:

c For a communications line

p For a parallel printer

s For a serial printer

Operation

After completing the command form, press GO. A parameter form for the device type you specified is displayed. Parameter values are described below.

Device-Type Parameter Forms

The parameter form for each device type contains fields that are relevant to that particular device. The fields for each device are described below.

```
Parallel Line Printer Parameters
  [New line mapping mode (binary, CR, or CR/LF)]
  [Tab expansion]
  [Number of characters per line]
  [Transmit time out (number of seconds)]
  [Receive time out (number of seconds)]
  [Additional ACK delay (units of 100 micro seconds)]
  [Translation file]
  [End of transmission character (none, CR, CR/LF, or FF)]

Serial Line Printer Parameters
  [Data bits (5, 6, 7, or 8)]
  [Parity (none, even, odd, 0, or 1)]
  [Baud rate]
  [Stop bits (1 or 2)]
  [Transmit time out (number of seconds)]
  [New line mapping mode (binary, CR, or CR/LF)]
  [Line control (none, XON/XOFF, CTS or both)]
  [Tab expansion]
  [Number of characters per line]
  [Translation file]
  [End of transmission character (none, CR, FF or CR/LF)]

Communications Parameters
  [Data bits (5, 6, 7, or 8)]
  [Parity (none, even, odd, 0, or 1)]
  [Transmit baud rate]
  [Receive baud rate (if different than transmit baud rate]
  [Stop bits (1 or 2)]
  [Transmit time out (number of seconds)]
  [Receive time out (number of seconds)]
  [CR/LF mapping mode (binary or new line)]
  [New line mapping mode (binary, CR, or CR/LF)]
  [Line control (none, XON/XOFF, CTS or both)]
  [EOF byte (hex value or none)]
  [NRZI option? (XC002 channels A and B only)]
  [Number of characters per line]
  [Tab expansion]
  [Ignore error type (break, framing, overrun, parity)]
```

Configurable Parameters

Communications devices, parallel printers, and serial printers use various combinations of the following parameter fields.

[Data bits (5, 6, 7, or 8)]

Default: 7

Enter the number of data bits per character. Data bits do not include the parity bit if parity is even, odd, 0, or 1.

[Parity (none, even, odd, 0)]

Default: 0

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]

Default: 9600

Enter the baud rate. Baud rate is a transmission speed from 20 to 19200 bits per second (bps). The transmit and receive baud rates are usually the same, except where the hardware allows the rates to be different. Common baud rates are 110, 150, 300, 1200, 2400, 4800, and 9600 bps.

[Transmit baud rate]

Default: 9600

Enter the baud rate. Baud rate is a transmission speed from 20 to 19200 bits per second. The transmit and receive baud rates are usually the same, except where the hardware allows the rates to be different. Common baud rates are 110, 150, 300, 1200, 2400, 4800, and 9600 bps.

[Receive baud rate]

Default: Same as transmit baud rate

Enter the baud rate. Baud rate is a receive speed from 20 to 19200 bps. The Receive and Transmit baud rates are usually the same, except where the hardware allows the rates to be different. Common baud rates are 110, 150, 300, 1200, 2400, 4800, and 9600 bps.

[Stop bits (1 or 2)]

Default: 1

Enter the number of stop bits per character.

[Transmit time out]

Default: Wait forever

Enter the number of seconds a write operation waits to begin transmitting a character before returning the following error code:

Device not ready (Error 300)

[Receive time out]

Default: Wait forever

Enter the number of seconds a read operation waits without receiving a character before returning the following error code:

No character available (Error 602)

If 0 is specified and no character is currently available, error code 602 is immediately returned from a read operation. If the default (wait forever) is used, error code 602 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 mapping of incoming carriage returns (CR) and line feeds (LF). Binary does not map any incoming CRs or LFs; it receives them unchanged. New Line maps an incoming single CR, single LF, or CR/LF combination into a carriage return (0Ah).

[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 (0Ah). Binary does not map any outgoing carriage returns; it transmits 0Ah. CR maps an outgoing carriage return into an ASCII CR (0Dh). CR/LF maps an outgoing carriage return into a carriage return/line feed sequence (ASCII CR/LF (0Dh/0Ah)).

[Line control (none, XON/XOFF, CTS, or both)]

Default: XON/XOFF

Enter **None**, **XON/XOFF**, **CTS**, or **Both**. This field determines the line control mode 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 hardware 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 (hex value or none)]

Default: 04

Enter the hexadecimal value of the byte to compare with the received byte to detect the end of the file, or enter **None**. If None is specified, the following error code is never returned from a communications bytestream.

```
End of file  (Error 1)
```

04 is the ASCII code for the EOT (End Of Transmission) character.

[NRZI option? (XC002 channels A and B only)]

Default: No

Enter **Yes** if the hardware you are using requires the NRZI (nonreturn-to-zero inverted) option.

[Number of characters per line]

Default: 132

Enter the maximum number of characters in a print line.

[Tab expansion]

Default: 8

Enter the number of blanks between tab characters.

[Additional ACK delay (units of 100 microseconds)]

Default: 0

For printers that use an acknowledgment signal (ACK) longer than 10 to 20 microseconds, enter the additional delay.

[Ignore error type (break, framing, overrun, parity)]

Default: None

Enter **B** (break), **F** (framing), **O** (overrun), or **P** (parity) to ignore that type of error. You can specify only one type of error.

[Translation file]

Default: None

Enter the file specification of the translation file you want to use. (See the **Make Translation File** command, later in this section, for a description of printer translation files.)

[End of transmission (none, CR, FF or CR/LF)]

Default: CR

Enter **None**, **CR**, **FF**, or **CR/LF**. This character or character sequence is appended to the end of the print buffer. If you change the default, the last print buffer might not be cleared, and remaining data can appear at the beginning of the next print job. This feature should be used only when printing with applications that clear the print buffer.

Default Configuration Files

Standard Software supplies default configuration files for use with printers and tape drives, as follows:

- Parallel printer

  [Sys]<Sys>LptConfig.sys and *[Sys]<Sys>SplConfig.sys*

- Serial printer

  [Sys]<Sys>PtrBConfig.sys and *[Sys]<Sys>SplBConfig.sys*

Table 3-2 lists the parameter values contained in the default printer configuration files. See "Configurable Parameters," above, for parameter descriptions.

Configuration Files for Communications Lines

Standard Software does not include default configuration files for communications lines. See the documentation for your communication device to determine appropriate parameter values.

The following file naming conventions are recommended when you create configuration files for communication devices:

ChannelName*Config.sys*

For example:

CommAConfig.sys

Communications channels on workstations are named as follows:

[Comm]A and *[Comm]B* on the processor module

[Comm]1A through *[Comm]1D* and *[Comm]2A* through *[Comm]2D* on the DCX and XC-002 Port Expander modules

Communications channels on shared resource processors are named as follows:

[Comm]A and *[Comm]B* on General Processors

[Comm]C through *[Comm]H* on the Communications Interface

[Comm]A through *[Comm]D* on Cluster Processors

[Comm]A through *[Comm]J* on Terminal Processors

Note: *Either or both Channel A and Channel B on workstations can be configured as communications lines. The default configuration file, which sets up Channel B for printing, is supplied for your convenience.*

Table 3-2. Create Configuration File Default Printer Parameters

Field Name	Parameter Value LptConfig.sys SplConfig.sys	 PtrBConfig.sys SplBConfig.sys
Data bits	Not used	7
Parity	Not used	Even
Baud rate	Not used	1200
Stop bits	Not used	1
Transmit timeout	Wait forever	Wait forever
Receive timeout	Wait forever	Not used
CR/LF mapping mode	Not used	Not used
New line mapping mode	CR/LF	CR/LF
Line control mode	Not used	XON/XOFF
EOF byte	Not used	Not used
Expand tab size	8	8
Number of characters per line	132	132
Additional ACK delay	0	Not used
Translation file	None	None
End of transmission	CR	CR

Create Directory

Use the **Create Directory** command to create a new directory on a CTOS volume.

Directory passwords and protection levels are also established with this command. See "File System Security," in Section 2, for more detailed information about passwords and protection levels.

This command is implemented with the Executive internal value !14.

Command Form

```
Create Directory
 New directory name(s)                    _____________________
 [Default protection level (15)]          _____________________
 [Maximum number of files (75)]           _____________________
 [Password for new directory]             _____________________
 [Volume password]                        _____________________
```

Parameter Fields

New directory name(s)

Enter the name(s) of the directory you want to create, using a maximum of 12 characters for each. Do not use braces ({ }), angle brackets (< >), square brackets ([ ]), slash (/), or backslash (\) in directory names.

If you are creating a directory on a volume other than the default, enter the volume or device name, as well as a name for the new directory, as shown in the following example:

[Volume]NewDir

Note that angle brackets are not required around the directory name.

[Default protection level (15)]

Default: 15

Enter a protection level for the directory. See Table 2-5 for a description of protection levels.

On real mode workstations, **Create Directory** accepts only protection level 0, 5, or 15 in this field. The protection level can be changed later, however, with the **Set Directory Protection** command.

[Maximum number of files (75)]

Default: 75

Enter the maximum number of files you want the directory to hold.

The number you enter in this field is only an approximation of the number of files the directory will hold. The actual number depends on the number of characters in each file name contained within the directory. **Create Directory** estimates 15 characters per file name.

Note: *Once a directory has been created, it cannot be enlarged to contain more files.*

[Password for new directory]

Default: No password

Enter a password for the new directory. Passwords can have up to 12 characters.

[Volume password]

Default: Current password

Enter the volume password, if required. Depending on how the system is configured, you may be able to create directories without the volume password.

Create File

Use the **Create File** command to create a new file of the size you specify. You seldom need to use this command, because most commands and applications create a new file automatically when you use a unique file specification for the first time. **Create File** is primarily used for creating swap files or extended crash dump files.

This command is implemented with the Executive internal value !5.

Command Form

```
Create File
 File name                          _____________________
 [Volume or directory password]     _____________________
 [File password]                    _____________________
 [File protection level]            _____________________
 [Size in sectors (0)]              _____________________
 [Overwrite ok?]                    _____________________
```

Parameter Fields

File name

Enter a file specification for the file you want to create.

[Volume or directory password]

Default: Currently active password

Enter the password for the volume or directory within which you want to create the file.

[File password]

Default: Currently active password

If you want to assign a password to the new file, enter it here. A file password is only valid if the file protection level supports the use of file passwords.

[File protection level]

Default: The same as the directory in which it is created

Enter the decimal value of the protection level for the new file. Only certain protection levels support the use of file passwords.

[Size in sectors (0)]

Default: 0

Enter the number of sectors you want to assign to the file. (See your Context Manager manual for information about swap file sizes; see the **Extended Crash Dump** command, later in this section, for information about extended crash dump file sizes.)

[Overwrite ok?]

Default: Ask for confirmation

If you enter **Yes** and the file already exists, the contents of the existing file are overwritten. If you enter **No** and the file already exists, the original file is preserved and the command is cancelled. If you leave the field blank, you are prompted for confirmation before the contents of the existing file are overwritten. Press **GO** to confirm, **CANCEL** to deny, or **FINISH** to terminate the command.

Create Keyboard Data Block

Use the **Create Keyboard Data Block** command to create a keyboard data block from a keyboard-table text file. Keyboard data blocks are used by programmers as follows:

- In object modules to be linked into the system keyboard table
- In object modules to be linked with specific applications
- As binary keyboard tables to be opened dynamically and used by one or more applications

The system keyboard table, *[Sys]<Sys>NlsKbd.sys*, is loaded into memory when the system bootstraps. It contains a data block for each standard Unisys keyboard, as well as for the PC-style keyboard. The loadable system keyboard table is supported by version 3.3 CTOS I and II (and subsequent higher versions). It replaces the *Nls.sys* and *Sys.keys* keyboard customization methods. A procedure for customizing the system keyboard table is included later in this section.

See "Customizing the System Keyboard Data Blocks," in the *CTOS Operating System Concepts Manual*, for information about customizing keyboard tables.

This command is implemented with the run file *KCT.run*, command case 00.

Command Form

```
Create Keyboard Data Block
  [Keyboard datafile name]          _____
  [Output is object module?]        _____
  [Output is binary file?]          _____
```

Parameter Fields

[Keyboard datafile name]

Default: See below

Enter the name of the keyboard-table text file that you want to convert. If you leave this field blank, an interactive screen is displayed, from which you can choose from a list of files ending with the suffix *.kbd_data* (see "Operation," below).

[Output is object module?]

Default: See below

Enter **Yes** to create an object module named Type*Kbd.obj* (where *Type* is the keyboard model, such as K1 or K5), that includes a keyboard data block. You may enter **Yes** in both this and the following field. If you enter **No** or leave this field blank, but specify **Yes** in the *[Output is binary file?]* field (see below), an object module is not created.

If you leave this and the following field blank, an interactive screen is displayed, from which you can choose the output-file type. See "Operation," below.

[Output is binary file?]

Default: See below

Enter **Yes** to create a binary keyboard data block named Type*Kbd.bin*. You may enter **Yes** in both this and the preceding field. If you enter **No** or leave this field blank, but specify **Yes** in the *[Output is object module?]* field (see above), a binary file is not created.

If you leave this and the preceding field blank, an interactive screen is displayed, from which you can choose the output-file type. See "Operation," below.

Operation

Depending on the parameters values you entered in the command form, one of several interactive screens is displayed. For example, if you left the command form blank, you are first prompted to choose an input file; or, if you specified object-module output, you are presented with an additional form containing object-module parameters.

The various interactive screens are described below.

Selecting a Text File

If you did not specify a text file name, a list of available text files ending with the suffix *.kbd_data* is displayed. From this display, you may select a file or search a different directory.

To select a file, position the highlight, then press **F7** (Select File).

To search a different directory, press **F3** (Change Dir), then **F4** (Confirm Dir).

Selecting Command Output

If you did not specify a type of output file (either object module or raw table) in the command form, the following function keys are now displayed. Press one to select a type of command output.

F1 (None)	Exits the command.
F2 (.obj Only)	Outputs an object module named *KnKbd.obj*, where *K*n is the keyboard model number (for example, K1, K5, and so on).
F3 (.tbl Only)	Outputs a binary keyboard table to a file named *KnKbd.tbl*.
F6 (.obj + .tbl)	Outputs both an object module and a binary keyboard table.

Specifying Output File Parameters

The following form is displayed:

```
Output file name                ____________________________
Type keyboard modified          ____________________________
IBus ID                         ____________________________
Include untranslate table?      ____________________________
PUBLIC in object module         ____________________________
```

To edit the form, press **F3** (Edit Above). Other function keys are **F1** (Exit NoSave), which exits the command without saving changes; **F5** (Only Trans), which creates a file with translation information only; **F7** (Only Emul), which creates a file with emulation information only; and **F9** (All Data), which creates a file using all data in the *.kbd_data* file.

Parameter fields are described below. Note that the last parameter appears only when you select object module output.

Output file name

This field is initially filled in with the default file name of the output type you selected (see the function key descriptions, above). If you want to change the default file name, enter the file name you want to assign to the file.

Type keyboard modified

This field is initially filled in with the current keyboard type. If this is not correct, enter the appropriate keyboard type; examples are K1, K2, K3, K4, or K5.

IBus ID

Enter the IBus identification number for the keyboard. This field is initially filled in with the IBus value for the current keyboard.

Include untranslate table?

Enter **Yes** to create a "untranslation table," for use when keyboard data is passed between two partitions that read different keyboard tables. (An untranslation table is large and adds considerably to the size of the keyboard table file. It does not, however, affect performance of the keyboard table, so file-size considerations are the only reason not to include it.)

PUBLIC in object module

This field appears only if you selected object module output. It is initially filled in with the string KeyBoardData. Enter the string you want to use as the PUBLIC name of the first byte of the table in the outputted object module.

Keyboard Customization Procedure

The Standard Software package contains an editable keyboard-table text file and an object module for each supported keyboard. You can use them to customize or internationalize the system keyboard table, as described below:

1. Edit the appropriate keyboard text file supplied with Standard Software or use one that you create with the **Convert Nls.sys** or **Convert Sys.keys** command.

   The keyboard text files supplied with Standard Software are named Type*Kbd.kbd_data*, where *Type* is the keyboard model, such as K1 or K5. Contents of these files are documented internally with comments inside the files.

2. Execute the **Create Keyboard Data Block** command on the customized text file, specifying **Yes** in the *[Output is object module?]* field, as shown in the following example:

```
Create Keyboard Data Block
  [Keyboard datafile name]     K2Kbd.kbd data
  [Output is object module?]   Yes
  [Output is binary file?]
```

3. Link the keyboard object module you just created with the object modules for other keyboards you want to support.

To be loadable by the operating system, the resulting keyboard table must be named *[Sys]<Sys>NlsKbd.sys*. See "Customizing the System Keyboard Data Blocks," in the *CTOS Operating System Concepts Manual*, for detailed information about the system keyboard table.

Create Message File

Use the **Create Message File** command to create binary message files, which applications use to display messages and prompts. This command functions as a compiler to convert ASCII message text files into a machine-readable binary format.

To customize or nationalize a message file, you edit the associated text file. After editing the text file, you create a new binary message file with the **Create Message File** command. For example, to modify screen messages displayed by the Executive, you edit the text file *ExecMsg.txt*. You then use **Create Message File** to create a new version of the binary message file, *ExecMsg.bin*.

See the Software Release Announcement for the operating system utilities for a list of message files provided with Standard Software.

This command is implemented with the run file *CreateMsgFile.run*.

Command Form

```
Create Message File
 Text file          _____________________________________
 [Message file]     _____________________________________
 [Print file]       _____________________________________
```

Parameter Fields

Text file

Enter the name of the text file containing the screen messages you want to change. (By convention, the names of these text files end with the suffix *.txt*, for example, *ExecMsg.txt*.)

[Message file]

Default: See below

Enter the name of binary message file you want to create or modify. (By convention, binary message files end with the suffix *.bin*.)If you leave this field blank, the default name is the first part of the text-file name with the suffix *.bin*. For example, if you specify *ExecMsg.txt* in the *Text file* field, the binary message file *ExecMsg.bin* is created. If the message file does not exist, it is created; if the message file already exists, its contents are overwritten (you are not prompted for confirmation).

It is a good idea to you save a copy of the original message file before executing this command.

[Print file]

Default: Screen only

If you want a record of screen output, enter a file or printer specification. If you leave this field blank, output is sent to the screen only.

Daily Archive

Use the **Daily Archive** command to perform an incremental backup to floppy diskettes, a hard disk, or tape. It combines the following tasks into a single command:

- Removes temporary files in the dollar-sign directories
- Optionally, erases the tape before the backup
- Optionally, disables the cluster if you are backing up the server, and resumes cluster operations when the backup is complete
- Performs an incremental backup of files modified since the most recent weekly backup
- Records the date and time of the backup in a log file

Before you can use this command to perform an incremental backup, you must perform a complete volume backup with the **Weekly Archive** command. See the documentation for that command, later in this section.

This command is implemented with the run file *SubCmd.run*, using the submit file *DailyArchive.sub*.

Command Form

```
Daily Archive
 Volume to archive                    _____
 [Volume password]                    _____
 [Disable cluster?]                   _____
 Archive dataset                      _____
 [Erase medium before archive?]       _____
```

Parameter Fields

Volume to archive

Enter the volume or device name of the disk you want to back up. It must match the volume or device name that you used with the **Weekly Archive** command. That is, if you used the device name for **Weekly Archive**, specify it again here. Likewise, if you used the volume name for **Weekly Archive**, specify it again here.

[Volume password]

Default: Active password

Enter the volume password of the disk you want to back up.

[Disable cluster?]

Default: No

If you are performing a backup of a disk on the server, and you want to disable the cluster, enter **Yes**. If you do not want to disable the cluster, leave this field blank.

Note: *Because* **Daily Archive** *is executed from a submit file, any response in this field disables the cluster. Therefore, you must leave this field blank if you do not want to disable the cluster.*

Archive dataset

To back up to tape, enter the device name of the tape drive you want to use. Surround the device name with square brackets, for example *[QIC]*.

To back up to disk, enter a file specification for the archive dataset, for example, *[f0]<Sys>Archive*.

[Erase medium before archive?]

Default: No

This parameter applies to tape backup media only. Enter **Yes** to erase the tape before the backup begins. If you do not want to erase the tape, leave this field blank.

Note that because **Daily Archive** is executed from a submit file, any response in this field erases the tape. Therefore, you must leave this field blank if you want to begin the backup without erasing the tape.

Method

Daily Archive is used in conjunction with the **Weekly Archive** command. They are designed to be used together as follows:

1. Use **Weekly Archive** to perform a complete volume backup. This command creates and maintains a date/time-stamp file named *[Sys]<Sys>*Volume*>ArchiveTime.txt* (where *Volume* is the volume or device name specified in the command form).
2. Use **Daily Archive** to perform an incremental backup of files that were created or modified since the weekly backup. **Daily Archive** reads the file *[Sys]<Sys>*Volume*>ArchiveTime.txt* to obtain the date and time for the incremental backup.

A record of the backup is written to *[Sys]<Sys>Archive.log*.

The **Daily Archive** command does not work until you have executed the **Weekly Archive** command on the disk you want to back up. In addition, and you must use the same convention, that is either the volume name or the device name, to identify the disk for both commands.

Deinstall CDROM Service

Note: *This command is not supported on EISA/ISA workstations.*

Use this command to deinstall the CD-ROM Service. **Deinstall CDROM Service** has no command form. To execute it, type the command name; then press **GO**.

It is implemented with the run file *DeinstallCDROMService.run*.

Deinstall MCR Service

Use this command to deinstall the Magnetic Card Reader Service. **Deinstall MCR Service** has no command form. To execute it, type the command name; then press GO.

This command is implemented with the run file *MCRInstall.run*, command case 01.

Deinstall Mouse Service

Use this command to deinstall the Mouse Service. **Deinstall Mouse Service** has no command form. To execute it, type the command name, then press GO.

This command is implemented with the run file *Mouse.run*, command case DM.

Deinstall Queue Manager

Note: *This command is not supported on EISA/ISA workstations.*

Use this command to deinstall the Queue Manager. **Deinstall Queue Manager** has no command form. To execute it, type the command name, then press **GO**.

This command is implemented with the run file *InstallQMgr.run*, command case DI.

Deinstall Screen Print

Use this command to deinstall the Screen Print Service. **Deinstall Screen Print** has no command form. To execute it, type the command name, then press GO.

This command is implemented with the run file *InstallScreenPrint.run*, command case DI.

Deinstall Sequential Access Service

Note: *This command is not supported on EISA/ISA workstations.*

Use this command to deinstall one or more tape drives from the Sequential Access Service and, after all tape drives have been deinstalled, the Sequential Access Service itself.

This command is implemented with the run file *InstallSeqService.run*, command case DI.

Command Form

```
Deinstall Sequential Access Service
  [Device(s) ([QIC])]
```

Parameter Fields

[Device(s) ([QIC])]

Default: *[QIC]*

Enter the name of the device or devices you want to deinstall from the Sequential Access Service. If more than one device is installed, each can be deinstalled individually. Only after the last device has been deinstalled is the Sequential Access Service itself deinstalled.

When a device is deinstalled, a message is displayed, as shown in the following example:

Device [QIC] has been deinstalled

When the Sequential Access Service is deinstalled, the following message is displayed.

The Sequential Access Service has been deinstalled

Deinstall Spooler

Note: *This command deinstalls the pre-Generic Print System (GPS) printing spooler only. Use the* **Print Manager** *command to deinstall GPS system services; see the* Generic Print System Administration Guide.

Use this command to deinstall the pre-GPS printing spooler. **Deinstall Spooler** has no command form. To execute it, type the command name, then press **GO**.

This command is implemented with the run file *Spooler.run*, command case DI.

Deinstall Statistics Service

Use this command to deinstall the Performance Statistics Service. **Deinstall Statistics Service** has no command form. To execute it, type the command name, then press GO.

This command is implemented with the run file *Statistics.run*, command case DI.

Deinstall Voice Service

Note: *This command is not supported on EISA/ISA workstations.*

Use this command to deinstall the Voice Service. **Deinstall Voice Service** has no command form. To execute it, type the command name, then press GO.

This command is implemented with the run file *InstallVoice.run*, command case DS.

Delete

Use the **Delete** command to remove files from a disk.

Note: *In some cases, deleted files can be recovered with the* **Undelete** *command. See the documentation for it, later in this section.*

This command is implemented with the Executive internal value !3.

Command Form

```
Delete
 File list
 [Confirm each?]
```

Parameter Fields

File list

Enter a file specification, a file list, or an at-file name. You can use wild cards in this field.

[Confirm each?]

Default: No

If you enter **Yes**, you are prompted to confirm the delete operation for each file by pressing GO.

If you enter **No** or leave the field blank, you are not prompted for confirmation, and all specified files are deleted.

Dir

The **Dir** command is identical to the **Files** command. Use it to display a list of file names, or optionally, to display detailed information about each file.

Dir is provided for users who are already familiar with DOS or UNIX. See the **Files** command, later in this section, for information about the command form, parameter fields, and display.

Disable Caching

The **Disable Caching** command pertains to the file system cache on protected mode processors. The file system cache is an area of memory where disk sectors are stored dynamically, as they are used. When sectors are cached, they are accessed from memory, rather than from disk. This increases the speed at which files are retrieved.

Use the **Disable Caching** command to change the caching attribute on specified files. A default attribute, either enable or disable caching, is specified for all files in the operating system configuration file (*[Sys]<Sys>Config.sys* on workstations, *[Sys]<Sys>SrpConfig.sys* on SRPs). In most cases, that default is set to enable caching. Therefore, you use the **Disable Caching** command to disable caching on selected files only.

See also "Using a File System Cache," in the *CTOS System Administration Guide*, and the **Enable Caching**, **Lock In Cache**, **Unlock Cache**, and **Cache Status** commands, in this section.

This command is implemented with the run file *CacheUtility.run*, command case 01.

Command Form

```
Disable Caching
 File list      _____________________________________
 [Print file]   _____________________________________
```

Parameter Fields

File list

Enter the file specifications of the files you do not want to cache.

[Print file]

Default: Screen only

If you want a record of screen output, enter a file or printer specification. If the file you specify does not exist, it is created. If it already exists, its contents are overwritten. If you leave this field blank, output is sent to the screen only.

Disable Cluster

Use the **Disable Cluster** command to disable cluster operations. It stops communications between the server and cluster workstations.

Disable Cluster can only be executed on the server. It is useful when you are installing or updating software in the cluster environment. You may also want to disable the cluster when you are backing up the server, to ensure that all files are included in the backup.

To resume cluster operations, use the **Resume Cluster** command, described later in this section.

This command is implemented with the run file *Cluster.run*, command case DI.

Command Form

```
Disable Cluster
  [Time interval in seconds]  _____
  [XE line to leave up]       _____
```

Parameter Fields

[Time interval in seconds]

Default: 0

Enter the number of seconds to delay before the cluster is disabled.

[XE line to leave up]

Enter the number of the SRP line to leave up so that cluster operations can be resumed. (If in doubt, use **Cluster Status** to determine your line number.)

Operation

After you execute the **Disable Cluster** command, the following message is displayed on cluster workstations:

```
Cluster going down ss
```

where *ss* is the number of seconds remaining until the cluster is disabled (*ss* is updated periodically during the interval).

After the cluster has been disabled, cluster workstations receive the following message when attempting to access the server:

```
Server workstation not running.  (Error 6)
```

Disk Partition Create

Note: *This command is supported on X-Bus+ and EISA/ISA-bus workstations only.*

Use the **Disk Partition Create** command to create a disk partition. For information about disk partitioning, see the **Format Disk** command, later in this section.

The function of this command is the same as that of the DOS **FDISK** command, and if the workstation is booted from DOS, you can use **FDISK**, rather than **Disk Partition Create**, to create DOS partitions.

This command is implemented with the run file *FormatDisk.run*, command case AP.

Command Form

```
Disk Partition Create
 Device name                          _____________________
 [Device or volume password]          _____________________
 [Partition type (CTOS)]              _____________________
 [Partition size in Mb]               _____________________
```

Parameter Fields

Device name

Enter the device name of the disk on which you want to create a partition.

[Device or volume password]

Default: No password

If the disk is a valid CTOS volume enter the volume password. If it is being used as a DOS drive only, enter the device password.

[Partition type (CTOS)]

Default: CTOS

To create a partition, enter the hexadecimal value from the table below that corresponds to the type of partition you want to create. You must specify an "h" after the value, as shown, for example, 06h.

Hexadecimal Value	Partition Type
0C0h	CTOS file system
0CDh	Crash dump
00h	Unused
01h	12-bit File Allocation Table (FAT)
04h	16-bit FAT
05h	Extended partition
06h	32-bit FAT
07h	OS/2 HPFS

Note that DOS partition types are operating-system version dependent. If you do not know what type of partition to specify, create your DOS partition with the **FDISK** command, after booting from DOS.

See the **Format Disk** command, later in this section, for more information about CTOS and crash-dump disk partitions.

[Partition size in Mb]

Default: All available disk space

Enter a size, in M bytes, for the partition you want to create.

Disk Partition Display Details

Note: *This command is supported on X-Bus+ and EISA/ISA-bus workstations only.*

Use the **Disk Partition Display Details** command to display a list of valid partitions on the specified disk.

This command is implemented with the run file *FormatDisk.run*, command case DP.

Command Form

```
Disk Partition Display Details
  Device name                    _____________________
  [Device or volume password]    _____________________
```

Parameter Fields

Device name

Enter the device name of the disk for which you want to display a list of partitions.

[Device or volume password]

Default: None

If the disk is a valid CTOS volume enter the volume password. If it is a DOS drive only, enter the device password.

Disk Partition Remove

Note: *This command is supported on X-Bus+ and EISA/ISA-bus workstations only.*

Use the **Disk Partition Remove** command to remove the most recently created partition.

This command is most frequently used to remove a DOS partition from a disk that contains both CTOS and DOS partitions. Note that removing a partition permanently destroys any data it contains. As a safeguard, however, you cannot remove an active partition.

> **Caution**
>
> This command permanently destroys data.

To remove the DOS partition, follow the steps below exactly. Failure to do so could destroy all data on the disk:

1. Bootstrap the workstation from the server or a floppy diskette.
2. Set CTOS as the active partition on the disk from which you want to remove the DOS partition. (See the **Disk Partition Set Active** command, later in this section.)
3. Reboot the workstation from the active CTOS partition.
4. Execute **Disk Partition Remove** to remove the DOS partition.

This command is implemented with the run file *FormatDisk.run*, command case VP.

Command Form

```
Disk Partition Display Details
 Device name                    _____________________
 [Device or volume password]    _____________________
```

Parameter Fields

Caution

Inadvertently specifying an incorrect device name in the following field, *Device name*, could destroy all data on that disk.

Device name

Enter the device name of the disk from which you want to remove a partition.

[Device or volume password]

Default: None

If the disk is a valid CTOS volume enter the volume password. If it is a DOS drive only, enter the device password.

Disk Partition Set Active

Note: This command is supported on X-Bus+ and EISA/ISA-bus workstations only.

Use the **Disk Partition Set Active** command to specify the partition from which the workstation is to bootstrap.

This command is usually used to specify a non-CTOS partition as the active partition. To reactivate the CTOS partition so that the workstation will boot from it, *c0* must be set as the active partition, using the software that is currently running on the workstation. For example, if the workstation is running DOS, the **FDISK** command is used to specify the active partition.

This command is implemented with the run file *FormatDisk.run*, command case SP.

Command Form

```
Disk Partition Set Active
 Device name                    _____________________
 [Device or volume password]    _____________________
 Partition ID                   _____________________
```

Parameter Fields

Device name

Enter the device name of the disk on which you want to set an active partition.

[Device or volume password]

Default: None

If the disk is a valid CTOS volume enter the volume password. If it is a DOS drive only, enter the device password.

Partition ID

Enter the ID number for the partition you want to be active. To find out the ID number, see the listing displayed by the **Disk Partition Display Details** command.

Disk Squash

Use the **Disk Squash** command to remove disk fragmentation without reformatting the disk. **Disk Squash** moves fragmented files and free sectors so that they become contiguous. This process improves file access speed and increases available disk space.

Before executing **Disk Squash**, it is recommended that you back up the disk with the **Volume Archive** command. See the documentation for it, later in this section.

The **Disk Squash** command does not work on CTOS III system volumes. Alternatives include booting from the server, making that the system volume, before executing **Disk Squash**; or backing up the disk, reinitializing it to remove fragmentation, and then restoring data from the backup. See the *CTOS System Administation Guide* for information about booting a workstation from the server and for disk backup and initialization procedures.

This command is implemented with the run file *Squash.run*.

Command Form

```
Disk Squash
 Device name                          _____________________
 [Volume password]                    _____________________
 [Display structures only?]           _____________________
 [Print file]                         _____________________
 [Suppress confirmation?]             _____________________
```

Parameter Fields

Device name

Enter the device name of the disk you want to squash.

[Volume password]

Default: None

Enter the volume password, if there is one.

[Display structures only?]

Default: No

Enter **Yes** if you want to display disk structure information but do not want to squash the disk. If you enter **No** or leave this field blank, fragmentation analysis appears on the screen, then restructuring of the volume begins. During restructuring, **Disk Squash** shows files being moved from one location on the disk to another.

Note: Fragmentation that remains after **Disk Squash** *has been run once can usually be corrected by running* **Disk Squash** *again.*

The fragmentation analysis is divided into the following parts:

Available sectors:

Range	Number of contiguous sector blocks
# Areas	Number of groups of a given range type
# Sectors	Total number of sectors of that range type

File allocation:

Range	Number of file extents
# Files	Number of files with a given file extent range

[Print file]

Default: Screen only

If you want a record of screen output, enter a file or printer specification. If the file you specify does not exist, it is created. If it already exists, current output is appended to the existing contents of the file.

If you leave this field blank, output is sent to the screen only.

[Suppress confirmation?]

Default: No

Enter **Yes** if you want **Disk Squash** to begin executing without further confirmation.

If you enter **No** or leave this field blank, you are given an opportunity to cancel **Disk Squash** before it begins executing.

Display Configuration

Use the **Display Configuration** command to display a picture of your workstation hardware configuration, including the amount of random access memory. The display also indicates if the workstation is booted from the server, rather than from its own local file system.

On workstations running a protected mode operating system, **Display Configuration** also shows the presence of a math coprocessor.

This command is implemented with the run file *DisplayConfig.run*.

Command Form

```
Display Configuration
  [Pause?]
```

Parameter Fields

[Pause?]

Default: No

This parameter applies to protected mode operating systems only. Specify **Yes** if you want the picture to remain on the screen until you press **FINISH** to exit the utility.

Dump

Use the **Dump** command to display the contents of a file in ASCII or hexadecimal format. You can also use **Dump** to compare and display the differences between two files.

This command is implemented with the run file *Dump.run*.

Command Form

```
Dump
  Input file                    _____________________________
  [Compare file]                _____________________________
  [Print file]                  _____________________________
  [File address first]          _____________________________
  [File address limit]          _____________________________
  [Continue on differences?]    _____________________________
  [EBCDIC?]                     _____________________________
```

Parameter Fields

Input file

Enter the name of the file you want to display.

[Compare file]

Default: None

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.

The comparison ends with one of the following messages:

```
Differences encountered.

No differences encountered.
```

[Print file]

Default: Screen only

Enter the name of a file to contain the comparison. If the file already exists, its contents are overwritten. If it does not exist, it is created.

[File address first]

Default: 0

Enter the hexadecimal number of the file address at which you want the comparison or display to begin. If the field is left blank or an invalid value is specified, the comparison or display begins at file address 0.

The expected value for this field is a hexadecimal number, so the conventional "h", which signifies that mode (for example, 1F2Eh), is optional.

[File address limit]

Default: FFFFFFFFh

Enter the hexadecimal number of the file address at which you want the comparison or display to end. The comparison or display is up to, but not including, this address.

If the number specified is greater than the end-of-file address, the comparison ends at the end of the file.

If you leave this field blank, or specify an invalid value, the comparison or display ends at file address FFFFFFFFh. The final "h", which indicates a hexadecimal number, is optional.

[Continue on differences?]

Default: No

If you enter **No** or leave this field blank, the comparison between two files is not continued if a difference is encountered.

If you enter **Yes**, the comparison continues, even if a difference is encountered.

[EBCDIC?]

Default: No

If you answer **Yes**, the right-hand side of the dump will be displayed using the Extended Binary Coded Decimal Interchange Code rather than the standard character set.

Editor

Use the **Editor** command to start the Editor, a text editing application designed for writing programming source code. The Editor can also be used to edit configuration files and to create simple documents. See the *CTOS Editor User's Guide* for complete information about the Editor application.

This command is implemented with the run file *Editor.run*.

Command Form

```
Editor
  [File name(s)]
  [Read only?]
  [Alternate user name]
  [Starting line number]
```

Enable Caching

The **Enable Caching** command pertains to the file system cache on protected mode processors. The file system cache is an area of memory where disk sectors are stored dynamically, as they are used. When sectors are cached, they are accessed from memory, rather than from disk. This increases the speed at which files are retrieved.

Use the **Enable Caching** command to change the caching attribute on specified files. A default attribute, either enable or disable caching, is specified in the operating system configuration file (*[Sys]<Sys>Config.sys* on workstations, *[Sys]<Sys>SrpConfig.sys* on SRPs). In most cases, that default is set to enable caching on all files. If, however, the default is set to disable caching, you use the **Enable Caching** command to enable caching on selected files only. You also use this command to reenable caching on previously disabled files.

See also "Using a File System Cache," in the *CTOS System Administration Guide*, and the **Disable Caching**, **Lock In Cache**, **Unlock Cache**, and **Cache Status** commands, in this section.

This command is implemented with the run file *CacheUtility.run*, command case 02.

Command Form

```
Enable Caching
 File list      _____________________________________
 [Print file]   _____________________________________
```

Parameter Fields

File list

Enter the name of the file(s) that are now permitted to be placed in the cache.

[Print file]

Default: Screen only

If you want a record of screen output, enter a file or printer specification. If the file you specify does not exist, it is created. If it already exists, its contents are overwritten. If you leave this field blank, output is sent to the screen only.

Expand File

Use the **Expand File** command to increase or decrease the size of a file, without reinitializing the disk. This is useful, for example, to increase the size of *<Sys>SysImage.sys* and other system files.

The **Expand File** command cannot be used on *<Sys>Mfd.sys* and *<Sys>FileHeaders.sys* on any volume, and on CTOS III system volumes, it also cannot used on *[Sys]<Sys>CrashDump.sys*, *[Sys]<Sys>Log.sys*, or *[Sys]<Sys>SysImage.sys*. On partitioned disks, **Expand File** cannot be used on *<Sys>CrashDump.sys*, *<Sys>MasterBootRecord.sys*, *<Sys>BootRecord.sys*, or *<Sys>DiskPartitions.sys*.

In addition, **Expand File** is supported on local disks only. If the Debugger is installed, deactivate it (**CODE-k**) before executing this command.

This command is implemented with the run file *Expander.run*, command case 10.

> **Caution**
>
> This command can destroy data when used to shrink the size of a file. In the case of *<Sys>SysImage.sys*, such data loss destroys the file from which the system bootstraps, which renders the system unbootable.

Command Form

```
Expand File
  [Device name ([D0])
  [Volume password]
  File name
  File size in sectors
  [Copy old to new?]
```

Parameter Fields

[Device name ([D0])

Default: *d0*

Enter the volume or device name of the disk on which you want to expand or shrink a file. Square brackets are not required.

[Volume password]

Default: Currently active password

Enter the volume password of the disk you specified.

File name

Enter the file specification of the file you want to expand or shrink, in the form of *<Directory>FileName*.

This command is frequently used to increase the size of *<Sys>SysImage.sys*, without reinitializing the disk. That allows you to update to a larger operating system or to use the disk as a system volume. Note, however, that after increasing the size of *SysImage.sys*, you must install the desired operating system. For example, after you have expanded *SysImage.sys* from 0 to 768 sectors, it remains an empty (although larger) file until you install the appropriate operating system.

File size in sectors

Enter a size for the file, in sectors.

[Copy old to new?]

Default: No

Specify **Yes** if you want to retain the contents of the file. If data in the existing file occupies more sectors than the newly specified size, existing data will be truncated.

Extended Crash Dump

Note: *This command is not supported on EISA/ISA workstations.*

Use the **Extended Crash Dump** command to dump the contents of protected-mode memory to a disk file on workstations that support extended crash dumping. See the *CTOS System Administration Guide* for detailed information about crash dumping and extended crash dumping on the various models of workstations and SRP processors.

Note: *A workstation can be set up so that the extended crash dump operation takes place automatically after a system crash; see "Collecting a Crash Dump," in the* CTOS System Administration Guide.

This command is implemented with the run file *ExtCrashDump.run*.

Command Form

```
Extended Crash Dump
  [Crash file]
  [Extended crash file]
```

Parameter Fields

[Crash file]

Default: *[Sys]<Sys>CrashDump.sys*

Enter the name of the file containing the memory dump for the first megabyte of memory, for example, *[Sys]<Sys>Ws>CrashDump.sys* on a diskless workstation.

[Extended crash file]

Default: *[Sys]<Sys>ExtCrashDump.sys*

Enter the file specification of the extended crash dump file. An extended crash dump file is not created when the system volume is initialized; it must be created with the **Create File** command.

The extended crash dump file must be large enough to contain the entire contents of memory. To determine the correct size, multiply the total amount of memory by two. For example, if the workstation has 2048K bytes of memory, allocate 4096 sectors when you create the extended crash dump file.

FdArchive

Use the **FdArchive** command to format floppy diskettes with parameters suitable for storing backups. Diskettes are initialized with the volume name *[Archive]*.

FdArchive has no command form. To execute it, insert a diskette into drive *[f0]*, type the command name, then press **GO**.

This command is implemented with the run file *SubCmd.run* and the submit file *FdArchive.sub*.

FExtent Reporter

Use the **FExtent Reporter** command to determine if it is necessary to reformat or squash a disk. **FExtent Reporter** reports the number of disk extents for a specified list of files. For best performance, a file should be one contiguous disk extent.

See also the **Disk Squash** command and the **Volume Checkerboard Reporter** command, in this section.

This command is implemented with the run file *Files.run*, command case 01.

Command Form

```
FExtent Reporter
  [File list (*)]
  [Suppress sort?]
  [Print file]
  [Summary only?]
```

Parameter Fields

[File list (*)]

Default: All files in the default path

Enter the file specifications to include in the file extent report.

[Suppress sort?]

Default: No

Enter **Yes** to suppress alphabetical sorting of the specified files. If you suppress sorting, the command executes faster.

If you enter **No** or leave this field blank, the file list is sorted alphabetically.

[Print file]

Default: Screen only

If you want a record of command output, enter a file or printer specification. If the file already exists, its contents are overwritten. If it does not exist, it is created. If you leave this field blank, command output is sent to the screen only.

[Summary only?]

Default: No

Enter **Yes** to display a summary of the file extent report, which consists of those files having the greatest number of extents.

If you enter **No** or leave this field blank, the report includes fragmentation details for each specified file.

Files

Use the **Files** command to display a list of file names on the screen. Output from the **Files** command can be written to a file, which you can use later as an at-file parameter value (see "At Files," in Section 2). Optionally, detailed information about a list of files can be displayed.

This command is implemented with the run file *Files.run*.

Command Form

```
Files
  [File list (*)]
  [Details?]
  [Print file]
  [Suppress sort?]
  [Max columns]
  [Sort by suffix?]
  [Suppress error messages?]
```

Parameter Fields

[File list (*)]

Default: All files in the default path

Enter a file specification defining the file or list of files you want to display. You can use the asterisk (*) and question mark (?) wild cards in this field.

You can also use the tilde (~), which is an exclusionary wild card for use with the **Files** command only. It is used to exclude part of a list of files. To use it, type the file specification you want to list, followed by a tilde, and then the characters you want to exclude, as shown in the following example:

```
[File list (*)] Section*  ~-old
```

In the above example, all files matching the first wild card pattern, *Section**, except those ending with *-old*, are displayed.

Note that the exclusionary wild card, when used by itself, means to exclude all files. Therefore, in the following example, * specifies all files, but then ~ excludes them, so nothing is displayed:

```
[File list (*)] *  ~
```

[Details?]

Default: No

If you enter **Yes**, the following information is displayed for each file:

- File length in characters
- Number of sectors
- Date last modified
- Protection level

In addition, the total number of files, sectors, and length in characters is displayed at the bottom of the screen, for the group of listed files.

You can display more or different information for this parameter by changing the file *ExecMsg.txt*. See the **Create Message File** command, in this section, and "Customizing Standard Software," in the *CTOS System Administration Guide*.

[Print file]

Default: Screen only

If you want a record of screen output, enter a file or printer specification. If the file you specify does not exist, it is created. If it already exists, its contents are overwritten. If you leave this field blank, output is sent to the screen only.

[Suppress sort?]

Default: No

Enter **Yes** to suppress alphabetical sorting of the specified files. If you suppress sorting, the command executes faster. (This is particularly useful when searching several large directories for a small group of files.) If you enter **No** or leave this field blank, the file list is sorted alphabetically.

[Max columns]

Default: See below

Enter the number of columns in which to display the command output. (If you specified a print file, this number applies to it, too.) If you leave the field blank, command output is displayed in a format that fits on the screen.

[Sort by suffix?]

Default: No

Enter **Yes** to sort file names alphabetically by suffix. The **Files** command identifies characters following a period (.) as a suffix. When you choose this option, files without suffixes are displayed first, followed by files grouped in alphabetical order, by suffix.

If you enter **No** or leave this field blank, files are alphabetized by first characters.

[Suppress error messages?]

Default: No

Enter **Yes** if you do not want error messages displayed in the command output. This is useful when you are using the *[Print file]* option and do not want errors, such as "No such directory," written to the file.

If you enter **No** or leave this field blank, errors generated by the **Files** command are displayed on the screen and written to the print file, if you specified one.

Floppy Copy

Use the **Floppy Copy** command to make duplicate copies of diskettes. You do not need a dual floppy-drive workstation to use this command.

The diskette you want to copy is called the master diskette; the diskette to which you are copying is called the destination diskette. **Floppy Copy** formats each destination diskette before it makes the copy; therefore, you can make copies onto blank, unformatted diskettes. Note, however, that **Floppy Copy** does not append data if the destination diskette already contains files. Any files that exist on a destination diskette are completely destroyed by this command.

Floppy Copy makes an exact duplicate of the master diskette. However, it does not duplicate diskettes that have unrecoverable bad spots. If a master diskette contains bad spots, create a new master diskette (using **Volume Archive**, **Format Disk**, and **Restore Archive**) before you begin making copies.

This command is implemented with the run file *FloppyCopy.run*.

Caution

This command destroys all existing data on the destination diskette.

Command Form

```
Floppy Copy
  [Number of copies]   _____
  [Overwrite ok?]      _____
  [Dual floppy?]       _____
  [Suppress verify?]   _____
  [Device name(s)]     _____
  [Volume 1 password]  _____
  [Volume 2 password]  _____
```

Parameter Fields

[Number of copies]

Default: 1

Enter the number of copies you want to make from each master diskette. After the specified number of copies has been made, you are prompted to insert the next master diskette.

[Overwrite ok?]

Default: No

If you enter **No** or leave this field blank, you are asked for confirmation to reformat a destination disk that contains data. (This alerts you that you are overwriting a valid floppy volume.)

If you enter **Yes**, the destination diskette is automatically formatted, whether or not it contains data.

[Dual floppy?]

Default: See below

If you are using a single floppy-drive system, the default for this field is **No**. If you enter **No** or leave this field blank, you are prompted to insert the master diskette into drive *[f0]*. If you enter **Yes**, an error message informs you that no device named *[f1]* exists on your workstation.

If you are using a dual floppy-drive system, the default for this field is **Yes**. If you enter **Yes** or leave this field blank, you are prompted to insert the master diskette into the first floppy drive specified in the *[Device name(s)]* field (described below) and a blank diskette into the second floppy drive named in the *[Device name(s)]* fields.

If you are using a dual floppy-drive system and enter **No**, only one drive is used, as specified in the *[Device name(s)]* field. The command then functions as described above for a single floppy-drive system.

[Suppress verify?]

Default: No

If you enter **No** or leave this field blank, the destination diskette is verified against the master diskette, after the copy operation is complete. If you enter **Yes**, the destination diskette is not verified.

[Device name(s)]

Default: See below

If you are using a single floppy-drive system, enter the device name of the floppy drive. The default is *[f0]*.

If you are using a dual floppy-drive system, you may enter two device names, separated by a space. The first device name is the drive to contain the master diskette; the second device name is the drive to contain the destination diskette. Defaults are *[f0]* and *[f1]*, in that order.

If you want to use only one drive on a dual floppy-drive system and that drive is not *[f0]*, enter the device name in this field. You must also specify **No** in the *[Dual floppy?]* field, as described above.

[Volume 1 password]

Default: No password

Enter the password for the master diskette.

[Volume 2 password]

Default: No password

Enter the password for the destination diskette.

Floppy Install

Use the **Floppy Install** command to install software applications from floppy diskettes. It is identical to invoking the **Floppy Install** option from the **Installation Manager** menu. See the **Installation Manager** command, later in this section, for information about installation procedures.

This command is implemented with the run file *InstallMgr.run*, command case IF.

Command Form

```
Floppy Install
  [Append to log file?]
```

Parameter Field

[Append to log file?]

Default: No

Enter **Yes** if you want to append to, rather than overwrite, an existing log file. If you enter **No** or leave this field blank, the installation log file is overwritten with each successive installation procedure.

Format

Use the **Format** command to format and print an ASCII text file as a paginated document. With this command, you can print to a spooled or direct printer, or to a file.

The defaults listed below produce text that fits an 8.5 by 11-inch page, if the printer prints six vertical lines and ten horizontal spaces per inch.

This command is implemented with 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 (10 spaces)]                  _____
 [Text width (65 spaces)]                   _____
 [Page length (66 lines)]                   _____
 [Top margin (6 lines)]                     _____
 [Bottom margin (6 lines)]                  _____
 [Tab width (8 spaces)]                     _____
 [Suppress page ejects between files?]      _____
```

Parameter Fields

File list

Enter the file specification, file list, or at-file name for the file(s) you want to format.

[Print to]

Default: *[Spl]*

Enter one of the following:

- For GPS and pre-GPS spooled printers, a queue name enclosed in square brackets, for example, *[SplB]* or *[Gps1]*
- For a GPS direct printer, a printer name enclosed in square brackets, for example, *[LaserPrinter]*
- For a pre-GPS direct printer, a device specification enclosed in square brackets, for example, *[Ptr]B*
- For a pre-GPS printer on another node, a node name, followed by a queue name, for example *{OtherNode}[SplB]*
- A file specification, for example, *[Sys]<Sys>PrintFile*

[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, the file is automatically formatted.

[Title]

Default: *FileName*

Enter a title to be printed at the top of each page. If the title includes spaces, enclose the title in single quotation marks ('). To prevent a default title from being printed, enter a pair of single quotation marks ('').

[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 **No** or leave this field blank, sequential page numbers, starting with 1, are printed at the bottom of each page. If you enter **Yes**, page numbers are not printed.

[Suppress date?]

Default: No

If you enter **No** or leave this field blank, the system date is printed at the top of each page. If you enter **Yes**, the date is not printed.

[Suppress time?]

Default: No

If you enter **No** or leave this field blank, the system time is printed at the top of each page. If you enter **Yes**, the time is not printed.

[Double-space?]

Default: No

If you enter **No** or leave the field blank, printed lines are single-spaced. If you enter **Yes**, printed lines are double-spaced.

[Left margin (10 spaces)]

Default: 10 spaces

Enter the number of spaces you want to allow for the left margin.

[Text width (65 spaces)]

Default: 65 spaces

Enter the number of characters you want to print on each line. Lines longer than the specified length are divided into two or more lines.

[Page length (66 lines)]

Default: 66 lines

Enter the number of lines per page, including the top and bottom margins.

[Top margin (6 lines)]

Default: 6 lines

Enter the number of blank lines you want to allow for the top margin.

[Bottom margin (6 lines)]

Default: 6 lines

Enter the number of blank lines you want to allow for the bottom margin.

[Tab width (8 spaces)]

Default: 8 spaces

Enter the number of spaces you want between tab stops.

[Suppress page ejects between files?]

Default: No

If you enter **No** or leave the field blank, each file is started on a new page. If you enter **Yes**, files are printed as continuous text.

Example

The following example formats three files into a single document:

```
Format
 File list                                a1 a2 a3
 [Print to]                               [Ptr]A
 [Confirm each?]
 [Title]                                  'Draft 3'
 [First page to format]
 [Last page to format]
 [Suppress page numbers?]
 [Suppress date?]
 [Suppress time?]                         Yes
 [Double-space?]                          Yes
 [Left margin (10 spaces)]
 [Text width (65 spaces)]
 [Page length (66 lines)]
 [Top margin (6 lines)]
 [Bottom margin (6 lines)]
 [Tab width (8 spaces)]
 [Suppress page ejects between files?]    Yes
```

A single document composed of the three input files is printed with double spacing as continuous text. The date and title are printed at the top of each page, and the page number at the bottom.

Format Disk

Use the **Format Disk** command to format disks and initilaize them with CTOS volume structures. In addition to command form parameters, the **Format Disk** command reads many of its parameter values from a configuration file, which can be customized and saved for specific configurations. The information in this manual concentrates on describing the command form and configuration file parameters. For detailed information about disk types and step-by-step procedures for formatting disks, see Section 11, "Adding Hard Disks," in the *CTOS System Administration Guide*.

On X-Bus+ and EISA/ISA-bus workstations, **Format Disk** can reserve disk space for non-CTOS partitions; see "Partitioned Disks," later in this section, for more information.

This command is implemented with the run file *FormatDisk.run*, command case FF.

> **Caution**
>
> This command destroys all data on the disk.

Command Form

```
Format Disk
  Device name                    _____
  [Device password]              _____
  [Current volume password]      _____
  [New volume name]              _____
  [New volume password]          _____
  [Configuration file]           _____
  [Format template]              _____
  [Device template]              _____
  [Print file]                   _____
  [Overwrite ok?]                _____
  [Bad spot file]                _____
  [Recalculate defaults?]        _____
  [CTOS partition size in Mb]    _____
```

Parameter Fields

Device name

Enter the device name of the disk you want to initialize.

[Device password]

Default: None

Enter the device password for the disk you are initializing.

Device passwords are assigned by the operating system. If you are using one of the released prebuilt operating systems, device passwords are the same as device names for hard disk drives (that is, *d0*, *d1*, and so on) and none for floppy drives. If you are using a customized operating system, it may assign different device passwords.

[Current volume password]

Default: Currently active password

If you are reinitializing a valid volume, enter the volume password currently assigned to it. When you are initializing a new disk, there is no volume password to enter here.

[New volume name]

Default: Current volume name

Enter a name, up to 12 characters long, to assign to the volume. It can contain letters, numerals, periods, and hyphens. If the disk is a valid volume, this parameter must be entered. If you leave this field blank, the current volume name is retained.

The following punctuation marks are reserved for use in file specifications and should not be used in volume names.

{ and *}* (braces)

< and > (angle brackets)

[and *]* (square brackets)

/ and \ (slash and backslash)

Other punctuation marks are also restricted; see "Restrictions for Volume, Directory, and File Names," in the *CTOS Executive User's Guide*. In addition, do not name a volume any of the following (see also "Device Specifications," in Section 2):

Comm*anything*	QIC*anything*
CTOS*anything*	Scr
d0, d1, d2, f0, f1	Spl*anything*
Kbd	Sys*anything*
Lpt*anything*	Tape*anything*
Nul	Vid*anything*

[New volume password]

Default: None

Enter a password, up to twelve characters long, to assign to the volume. It can contain letters, numerals, periods, and hyphens.

If you do not assign a volume password, file system security mechanisms are not implemented. See "Implementing System Security," in the *CTOS System Administration Guide*.

[Configuration file]

Default: *[Sys]<Sys>FormatDiskConfig.sys*

Enter the name of the configuration file containing the format and device templates to be used for formatting the disk. (See also "Parameter Templates," later in this section.)

[Format template]

Default: See below

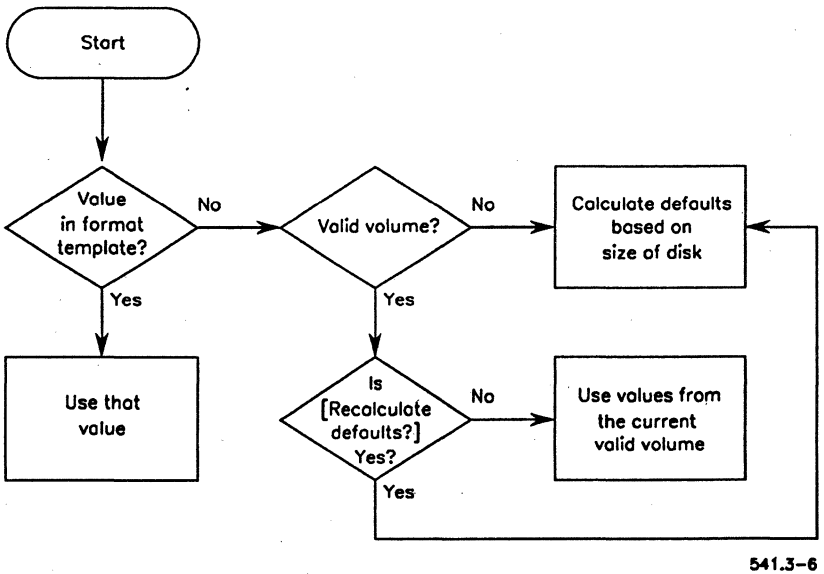
Enter the name of a format template containing volume parameters for the disk. See Table 3-3, later in this section, for a list of standard format templates.

If you leave this field blank, the disk is initialized with default values, as follows:

1. If the disk is a valid volume, it is reinitialized with existing parameters.
2. If the disk is a valid volume, and you specify **Yes** in *[Recalculate defaults?]* (described below), it is reinitialized with parameters calculated by **Format Disk**. These are based on the size of the disk.
3. If the disk is not a valid volume, it is initialized with parameters calculated by **Format Disk**.

See Figure 3-6 for a diagram and "Parameter Templates," later in this section, for more detailed information.

Figure 3-6. Format Disk Volume Default Sequence



[Device template]

Default: See below

For SCSI disks and EISA/ISA IDE drives, leave this field blank. Device parameters are taken directly from the hardware and any parameters specified in a device template will be ignored.

For other disks, enter the name of a device template containing the physical device characteristics of the disk; or, if that is unknown, enter the actual device characteristics in the form of *Cylinders/Heads/SectorsPer Track*. See Table 3-4 for a list of standard device templates. If you leave this field blank, **Format Disk** attempts to format the disk with default parameters. In some cases, however, **Format Disk** fails if you leave this field blank. Also, be aware that the default parameters might not be optimal for the disk.

If you are reinitializing a valid volume, you can leave this field blank.

[Print file]

Default: Screen only

If you want a record of command output, enter a file or printer specification. If you leave this field blank, command output is sent to the screen only.

[Overwrite ok?]

Default: Ask for confirmation

Enter **Yes** if you are aware that the disk is a valid volume and you want to reinitialize it. If you enter **No**, and the disk is a valid volume, **Format Disk** exits without reinitializing the disk. If you leave this field blank, and the disk is a valid volume, you are prompted to press GO to reinitialize the disk.

[Bad spot file]

Default: None

Enter the file specification of the file containing bad spot information for the disk. In many cases, this parameter is not required for SCSI disks, EISA/ISA IDE drives, or previously formatted disks. See "Bad Spots Files," later in this section. See also "Correcting Input/Output Errors," in the *CTOS System Administration Guide*, for step-by-step procedures.

[Recalculate defaults?]

Default: No

This field applies to volumes that are already initialized, if a specific format template is not specified. Enter **Yes** to reinitialize the disk with defaults calculated internally by the **Format Disk** command. (If you specified a format template in the *[Format template]* field, a **Yes** response in this field is ignored, and parameters are taken from the format template.)

If you enter **No** or leave this field blank, the disk is reinitialized with its current parameters.

[CTOS partition size in Mb]

Default: All available disk space

This parameter applies to X-Bus+ and EISA/ISA-bus workstations only.

Enter a size in M bytes for the CTOS partition. This reserves remaining disk space for later use as a non-CTOS partition. For example, to reserve 40M bytes of a 100M byte disk for DOS, enter 60 (for the CTOS partition) in this field. That leaves 40M bytes, on which you can later create a DOS partition. (See the **Disk Partition Create** command, earlier in this section, or the **FDISK** command, in your DOS documentation, for information about creating DOS partitions.)

If the disk is to be used exclusively as a non-CTOS drive, you can suppress the CTOS partition entirely. See the *:SuppressVolumeStructures:* parameter, under "Format Template Keywords," later in this section.

Parameter Templates

The **Format Disk** command reads parameter values from the configuration file you specify in the command form. Parameter values are grouped into templates within the configuration file. Format templates contain volume parameters, such as the maximum number of directories and files on the volume. Device templates contain disk hardware parameters. Examples are shown below.

Example Format Template

```
:FormatTemplate:DataVolume
:MaxFilesOnVolume:
:PrimaryFileHeadersOnly?:
:MaxDirectories:
:MaxFilesInSysDirectory:
:PasswordEncryption?:
:Debug?:Yes
:SysDirectoryPassword:
:ProtectSysDirectory?:
:SystemImageSize:0
:CrashFileSize:0
:SystemLogFileSize:0
:SuppressFormat?:Yes
:SurfaceTestsIfUnformatted:4
:SurfaceTestsIfFormatted:1
:OldCTOSFormat?:No
```

Example Device Template

```
:DeviceTemplate:Micropolis85
:CylindersPerDisk:1024
:TracksPerCylinder:8
:SectorsPerTrack:17
:BytesPerSector:512
```

The format templates supplied with Standard Software are listed in Table 3-3. To use a particular format template, specify its name in the *[Format template]* field of the **Format Disk** command form. The device templates supplied with Standard Software are listed in Table 3-4. To use a particular device template, specify its name in the *[Device template]* field of the **Format Disk** command form.

Table 3-3. Format Disk Standard Format Templates

Template Name	Characteristics
Defaults (if no format template is specified)	
FloppyDefault	Floppy diskette primary file headers only
DiskDefault	System volume without password protection
Floppy Volumes	
FloppyArchive	Floppy diskette suitable for backups
FloppyData	Floppy diskette with approximately 60 file headers
FloppyProtect	Floppy diskette with password protection and a write-protected *<Sys>* directory
X-Bus+ and EISA/ISA System Volumes	
SGenSysVolume	X-Bus+ or EISA/ISA system volume without password protection
SGenSysVolumeProtect	X-Bus+ or EISA/ISA system volume with password protection and a write-protected *<Sys>* directory
SGenSysVolumeEncrypt	X-Bus+ or EISA/ISA system volume with encrypted password protection and a write-protected <Sys> directory
SGenSavePartition	X-Bus+ or EISA/ISA system volume with an existing non-CTOS partition that you do not want to destroy
SGenSavePartitionProtect	X-Bus+ or EISA/ISA system volume with password protection, a write-protected *<Sys>* directory, and an existing non-CTOS partition that you do not want to destroy
SGenSavePartitionEncrypt	X-Bus+ or EISA/ISA system volume with encrypted password protection, a write-protected *<Sys>* directory, and an existing non-CTOS partition that you do not want to destroy

continued

Table 3-3. Format Disk Standard Format Templates (cont.)

Template Name	Characteristics
X-Bus Workstation System Volumes	
WSSysVolume	Workstation system volume without password protection
WSSysVolumeProtect	Workstation system volume with password protection and a write-protected *<Sys>* directory
WSSysVolumeEncrypt	Workstation system volume with encrypted password protection and a write-protected *<Sys>* directory
SRP System Volumes	
SrpSCSISysVolume	SRP SCSI system volume without password protection
SrpSCSISysVolumeProtect	SRP SCSI system volume with password protection and a write-protected *<Sys>* directory
SrpSCSISysVolumeEncrypt	SRP SCSI system volume with encrypted password protection and a write-protected *<Sys>* directory
SrpSMDSysVolume	SRP SMD system volume without password protection
SrpSMDSysVolumeProtect	SRP SMD system volume with password protection and a write-protected *<Sys>* directory
SrpSMDSysVolumeEncrypt	SRP SMD system volume with encrypted password protection and a write-protected *<Sys>* directory
SrpWinSysVolume	SRP ST-506 system volume without password protection
SrpWinSysVolumeProtect	SRP ST-506 system volume with password protection and a write-protected *<Sys>* directory
SrpWinSysVolumeEncrypt	SRP ST-506 system volume with encrypted password protection and a write-protected *<Sys>* directory

continued

Table 3-3. Format Disk Standard Format Templates (cont.)

Template Name	Characteristics
Data Storage Volumes	
DataVolume	Data storage only (a non-system volume) without password protection (workstation or SRP)
SmallDataVolume	Data storage only (a non-system volume) without password protection and approximately 500 file headers (workstation or SRP)
SrpDataVolume	Data storage only (a non-system volume) without password protection (SRP only)
Memory Disks	
MemDisk1	1-megabyte SRP memory disk
MemDisk3	3-megabytes SRP memory disk

Table 3-4. Format Disk Standard Device Templates

Manufacturer and Size of Disk	Device Template Name
Workstation disk vendor codes	A to Z and A1 to Z1
Atasi, 46 megabytes	Atasi46
Ball, 100 megabytes	Ball100
Control Data, 300 megabytes	CDC300
Control Data, 340 megabytes	CDC340
Control Data, 675 megabytes	CDC675
Fujitsu, 80 megabytes	Fujitsu80
Hitachi, 51 megabytes	Hitachi51
Hitachi, 85 megabytes	Hitachi85
Maxtor, 53 megabytes	Maxtor53
Maxtor, 143 megabytes	Maxtor143
Memorex, 166 megabytes	Memorex166
Micropolis, 52 megabytes	Micropolis52
Micropolis, 85 megabytes	Micropolis85
Nortel, 350 megabytes	Nortel350
Toshiba, 85 megabytes	Toshiba85
Memory disk, 1 megabyte	MemDisk1
Memory disk, 3 megabytes	MemDisk3
Regular capacity floppy disk	FloppyType
High-capacity floppy disk	FloppyTypeHiCap

Configuration File Format

If the standard templates are not suitable, you can edit or create a configuration file to add templates of your own. To do so, you use the Editor application to enter keywords and values.

Each line of the configuration file is written in the following format:

:Keyword:Value

where

:Keyword:	Identifies a parameter; keywords and the placement of colons must not be changed.
Value	Is a parameter value.

Format and device template keywords are described below. See "Using Parameter Templates," in the *CTOS System Administration Guide* for information about creating and editing parameter templates.

Format Template Keywords

The following parameters are included in format templates. The *:FormatTemplate:* keyword must be the first parameter in a format template, and you must assign it a unique value. The other keywords may appear in any order. If a keyword is omitted or a value left blank, a default value is used. If **Yes** is specified in the *[Recalculate defaults?]* field in the command form, defaults are determined as if the disk is unformatted.

:FormatTemplate:

Specify a unique name for the format template. This must be the first keyword in the format template and a value must be specified.

:MaxFilesOnVolume:

Default for valid volume: Current value
Default for unformatted disk: See below

Specify the maximum number of files to be stored on the volume. The maximum number of files is based on twenty-five sectors per file; therefore, the number you specify here may be adjusted by **Format Disk** according to the size of the disk.

To specify an exact number, precede this value with a pound sign (#), for example, **#5000**. When the value is preceded by a pound sign, **Format Disk** attempts to create the exact number of file headers specified; in cases where secondary file headers are requested, however, **Format Disk** creates additional file headers. If the number of files requested seems too large for the size of the disk, **Format Disk** reduces the number of files to a reasonable value.

If you do not specify a value, **Format Disk** calculates a value based on the size of the disk.

:PrimaryFileHeadersOnly?:

Default: No

Enter **Yes** if you want only primary file headers on the volume. This option is generally used for floppy diskettes only. If you enter **No** or leave this field blank, secondary file headers are created. These are needed to recover data from a corrupted volume.

:MaxDirectories:

Default for valid volume: Current value
Default for unformatted disk: See below

Specify the maximum number of directories to be created on the volume. If you do not specify a value, **Format Disk** calculates a value based on the size of the disk.

:MaxFilesInSysDirectory:

Default for valid volume: Current value
Default for unformatted disk: See below

Specify the maximum number of files to be stored in the <*Sys*> directory. For system volumes, 1000 to 2500 files are recommended. For other volumes, it is at your discretion.

If you do not specify a value, **Format Disk** calculates a value based on the size of the disk.

:PasswordEncryption?:

Default: No

Enter **Yes** if you want the password to be encrypted. This provides security against sophisticated users who might be able to "peek" at the volume password. Enter **No** if you do not want the password to be encrypted.

Note: *Encrypted passwords cannot be deciphered. Therefore, it is essential to keep accurate records of assigned passwords.*

:Debug?:

Default: No

Enter **Yes** to display an F for each track that is formatted (non-SCSI only) and a T for each track that is surface tested.

Enter **No** to suppress this information.

:SysDirectoryPassword:

Default for valid volume: Current value
Default for unformatted disk: None

Specify a password, up to twelve characters long, to assign to the <*Sys*> directory.

:ProtectSysDirectory?:

Default: No

Enter **Yes** to set the protection level of <*Sys*> to 5, which prevents users from changing or adding files without a password. See also "Implementing System Security," in the *CTOS System Administration Guide*.

Enter **No** to set <*Sys*> directory protection to 15, which does not require a password to change or create files.

:SystemImageSize:

Default for valid volume: Current value
Default for unformatted disk: 0

For nonsystem volumes, specify **0**. For system volumes, specify the number of sectors for *[Sys]<Sys>SysImage.sys*, the operating system file. For prebuilt operating systems, 1024 is recommended. If you are using a customized operating system, you may need a larger value.

:CrashFileSize:

Default for valid volume: Current value
Default for unformatted disk: 0

For nonsystem volumes, specify **0**.

For system volumes, specify the number of sectors required for *[Sys]<Sys>CrashDump.sys*, the crash dump file. To calculate this number for X-Bus and X-Bus+ workstations, multiply the amount of processor memory, up to 1024K bytes, by two. For EISA/ISA workstations, multiply the full amount of processor memory by two. For example, to create a crash file large enough to accommodate 4096K bytes of memory, specify **8192**.

On X-Bus+ and EISA/ISA workstations, **Format Disk** creates a partition of the size specified and maps that disk space to the crash dump file, *[Sys]<Sys>CrashDump.sys*. This is an internal difference only, however, and has no effect on workstation operations.

:SystemLogFileSize:

Default for valid volume: Current value
Default for unformatted disk: 0

For nonsystem volumes, specify **0**. For system volumes, enter a number of sectors for *[Sys]<Sys>Log.sys*, the system log file. To track system problems accurately, 48 sectors is recommended. If your system volume is small, however, you may specify a smaller value (for example, 10 to 20 sectors).

:SuppressFormat?:

Default: No

Enter **No** if the disk has never been formatted or if diagnostics have been run on it. If you are reinitializing a valid CTOS volume, enter **Yes**; this bypasses device-level formatting and decreases the amount of time it takes to reinitialize the disk. If you enter **Yes**, and the disk is not a valid volume, the **Format Disk** command fails with an I/O error (Error 301); change this value to **No** before reexecuting **Format Disk**.

On a partitioned disk, specify **Yes** in this field to reinitialize the CTOS partition, while preserving data in the non-CTOS partition. If you specify **No** or leave this field blank, all partitions on the disk are destroyed.

:SurfaceTestsIfUnformatted:

Default: 4

Specify the number of surface tests to run on an unformatted disk (a new disk or after disk diagnostics). It is not unusual for a disk to have bad spots, however, it is important that they are detected. When a bad spot is encountered during surface testing, no data will be stored on it.

Surface tests should be run in multiples of four so that every bit combination can be tested.

:SurfaceTestsIfFormatted:

Default: 1

Specify the number of surface tests to run on a formatted disk.

:OldCTOSFormat?:

Default: No

Enter **No** to format the disk such that it describes a logical block address (LBA) device. Such a disk is formatted optimally for its device characteristics and is for use with CTOS III 1.0, CTOS II 3.3, CTOS I 3.3, or CTOS/XE 3.0 and subsequent higher versions only.

If you enter **Yes**, the disk is formatted for backward compatibility with earlier versions of the operating system, specifically, versions earlier than CTOS I or II 3.3 and CTOS/XE 3.0. With SCSI disks, however, such backward compatibility may result in disks that are not formatted to full capacity, and that run more slowly than is optimal.

For floppy disks, it is recommended that you specify **Yes** to this parameter.

:Verify?:

Default: No

Enter **Yes** to verify that the disk has been formatted and initialized correctly. Enter **No** to bypass the verification phase.

:SuppressDefaultScsiPages?:

Default: No

Enter **Yes** to ensure that a SCSI disk is set to default device parameters before formatting. If you enter **No** or leave this field blank, the **Format Disk** command may fail on unformatted SCSI disks.

:SuppressVolumeStructures?:

Default: No

Enter **Yes** if you do not want **Format Disk** to create CTOS volume structures. This reserves the entire disk for future use as a non-CTOS partition. (After the disk has been formatted with the **Format Disk** command, use the DOS **FDISK** command to create and activate a DOS partition, then use the DOS **FORMAT** command to create the DOS file system. See your DOS documentation for information about **FDISK** and **FORMAT** commands.)

Device Template Keywords

Keywords and parameter values for device templates are listed below. They are included here in case you need to create a new device template. *Do not change parameters in the device templates supplied with Standard Software.* To determine parameter values for a new device template, see the documentation for the disk.

:DeviceTemplate:

This must be the first keyword in each device template. For its value, enter a unique device-template name. A unique name is one that does not duplicate the name of any other device template in the configuration file.

:CylindersPerDisk:

Default: 306

Enter the number of cylinders to allocate on the disk.

:TracksPerCylinder:

Default: 4

Enter the number of tracks to format for each cylinder.

:SectorsPerTrack:

Default: 16

Enter the number of sectors to format for each track. On SRP disks, you can increase usable disk space by changing this number to 17. This is the only parameter that should be changed in any of the device templates supplied with Standard Software.

:BytesPerSector:

Default: 512

Enter the number of bytes to be allocated in each sector.

:WritePreCompCyl:

Default: 0

Enter the number of the cylinder at which write-precompensation begins. Write-precompensation reduces I/O errors on inner cylinders of the disk. The default, 0, specifies no write-precompensation and is used on disks that do not support this feature.

:SeekStepRate:

Default: 0

Enter a number to represent the time interval in microseconds between successive step pulses when a seek command is issued. (See the Western Digital WD-2010 documentation.) Common values are 0 (the default) and 14.

:UtilizeEcc?:

Default: See below

Enter **Yes** to use error checking and correction (ECC) format.

The default is set according to the capabilities of the hardware. If you specify **Yes**, but ECC capability is not present, the disk is formatted in cyclic redundancy check (CRC) mode. Specify **No** if ECC capability is present but you do not want to use it.

:SpiralFactor:

Default: 0

Specify the sector offset from track to track on the disk. This field applies only to SMD disks.

:Removable?:

Default: No

Enter **Yes** if the disk is a removable storage medium.

:HiCapacityFloppy?:

Default: No

Enter **Yes** to format high capacity diskettes on a high capacity drive. Enter **No** to format regular capacity diskettes on a high capacity drive. Be sure to use the correct type of diskette for your drive; see the *CTOS Media User's Guide.*

This field is ignored by regular capacity floppy drives.

Bad Spots Files

A bad spots file is required by the **Format Disk** command for uninitialized non-SCSI disks, other than EISA/ISA IDE drives. When reinitializing a valid volume, a bad spots file is required to add bad spots that have occurred since the last time the disk was initialized.

To create a bad spots file, open a new file with the Editor application on a disk other than the one you are initializing. A floppy diskette may be used for this purpose. Then, type in the appropriate bad spot information. Bad-spot entry formats for non-SCSI and SCSI disks are described below.

A sample bad-spots file is shown in Figure 3-7. Leave a space between each bad-spot entry, as shown. Press **RETURN** at the end of each line; bad-spot entries should not wrap from one line to the next.

Figure 3-7. Format Disk Bad Spots File

```
981/0/#1 877/2/#0 877/2/#1 975/4/#15 969/5/#4 757/6/#15
943/1/#5 943/1/#6 415/4/#15 1000/5/#0 24/2/#0 425/4/#9
880/5/#5 1005/5/#8 1005/5/#9 32/2/#8 737/4/#12 924/5/#8
924/5/#9 494/6/#1 849/2/#12 849/2/#13 885/4/#4 941/5/#9
941/5/#10 524/6/#11 995/4/#10 995/4/#11
```

Non-SCSI Bad Spot Formats

For non-SCSI disks (except EISA/ISA IDE drives, which do not require a bad spots file), enter bad spots in one of the following formats. See the documentation for the disk to determine which one to use.

c/h/sb/bc

c/h/#sector

c/h/$sector

where

c	Is the cylinder number
h	Is the head number
sb	Is the starting byte number
bc	Is the number of bits in error
#sector	Is the sector number of a 512-byte sector
$sector	Is the sector number of a 256-byte sector

The form *c/h* declares all sectors on cylinder *c* and head *h* bad. For example, 3/0 declares all sectors on cylinder 3, head 0, as bad.

The form *c/h/#sector* declares sector number *#sector* on cylinder *c*, head *h*, as bad. For example, 3/0/#2 declares sector 2 on cylinder 3, head 0 as bad.

The form *c/h/$sector* declares sector number *$sector* on cylinder *c* head *h*, as bad. For example, 3/0/$4 declares sector 4 on cylinder 3, head 0 as bad.

Numbers for the first three formats can be either decimal or hexadecimal. For hexadecimal numbers, append the letter "h" to the end of the number. For example, 3/0/Ch is the same as 3/0/12.

For non-SCSI disks that are not valid volumes enter the bad spots listed in the manufacturer's bad spot report (such a report is shipped with non-SCSI hard disks). When reinitializing a valid volume, you need only enter bad spots that have been reported in the system error log. Therefore, unless you are aware of new bad spots, you do not need a bad spots file when reinitializing a valid volume.

SCSI Bad Spot Format

For SCSI disks, enter bad spots as reported in the system error log. (You can view or print the system error log with the **PLog** command.) SCSI bad spot numbers are listed as integers with 2 to 9 digits, for example, 24339.

A SCSI bad-spots file should contain only those bad spots reported by **PLog** since the last time the disk was initialized. Manufacturer's bad spots and those you have previously specified during a volume initialization are already known to the SCSI disk. Specifying them a subsequent time causes loss of usable disk space.

Partitioned Disks

On X-Bus+ and EISA/ISA-bus workstations, **Format Disk** partitions disk space. For use with CTOS, this is simply an internal difference that is of little or no consequence to users. If, however, you plan to boot DOS on your X-Bus+ or EISA/ISA-bus workstation, you will need to understand disk partitioning so that you can prepare the disk correctly.

On X-Bus+ and EISA/ISA workstations, **Format Disk** creates either one or two partitions on a hard disk. It creates a CTOS partition, which contains the CTOS file system and, if you specified a size for the crash dump file, it creates a crash-dump partition. It creates the CTOS partition to the size specified in the *[CTOS partition size in Mb]* field of the command form. Disk space in excess of what you specify for the CTOS partition is reserved; you can later partition and format that disk space for use with DOS.

Caution

Note that you can format a DOS partition with the DOS **FORMAT** command without affecting data in a CTOS partition. Likewise, you can initialize a CTOS partition with **Format Disk** without affecting a DOS partition if and only if you specify **Yes** to the *:SuppressFormat?:* parameter in the format template. If you fail to do so, **Format Disk** performs device-level formatting and all existing partitions are destroyed.

If you plan to run DOS on your X-Bus+ or EISA/ISA workstation, (either alternately with CTOS or exclusively), prepare the disk as follows:

1. If you will be using DOS exclusively, specify **Yes** to the *:SuppressVolumeStructures:* parameter in the format template you plan to use.
2. Reserve disk space for the DOS partition by specifying a size for the CTOS partition in the command form (see the *[CTOS partition size in Mb]* parameter, above).
3. After the disk is formatted, use the DOS **FDISK** command (preferred) or the CTOS **Disk Partition Create** command to create a DOS partition. (To use **FDISK**, boot the workstation from a DOS diskette.)
4. Format the partition with the DOS **FORMAT** command.

Head Park

Use the **Head Park** command to lock workstation disk drive heads away from the disk mounted in drive *[d0]*. This prevents the drive heads from damaging data on the disk surface when you relocate a workstation.

The **Head Park** command has no command form. To execute it, type the command name, then press **GO**. When the command finishes executing, turn the workstation off. After you move the workstation and turn it back on again, the heads are automatically unlocked and the disk is ready to use.

This command is implemented with the run file *HeadPark.run*.

Histogram

Use the **Histogram** command to obtain information about the proportion of time a protected mode program spends in a given process. The **Histogram** command reads the code segment and instruction pointer (CS:IP) at selected intervals and records the results. You specify whether to display information on all task segment selectors (TSSs) in the program, on selected TSSs, on all procedures, or on selected procedures.

This command is implemented with the run file *Histogram.run*.

Command Form

```
Histogram
  [TSSs to select (all)]  _____________________________
  [Symbol file]           _____________________________
  [Report by proc?]       _____________________________
  [Symbol file offset]    _____________________________
  [Print file]            _____________________________
  [Append to print file?] _____________________________
  [Timer interval (50 ms)] ____________________________
```

Parameter Fields

[TSSs to select]

Default: All

Enter the selected TSSs you want included in the histogram. Since there is a one-to-one correspondence between a process and a TSS, you can first enter the debugger, and then type CODE-S to display each process and its associated TSS.

[Symbol file]

Default: None

Enter the name of the symbol file if you want to include information by procedure name (symbol) as well as by address.

[Report by proc?]

Default: No

By default, information is reported by CS:IP only. Enter **Yes** if you want procedure names displayed, in addition to the CS:IP for each procedure. See "Options Form," later in this command description, for examples.

[Symbol file offset]

Default: 0

This field is required only for system services and for operating system symbols. Applications compute this offset automatically.

Enter the symbol file offset. To determine the symbol file offset, follow these steps:

1. Use the Debugger to examine the procedure or system service.
2. Look at the DS (data segment) register, for example:

   15D0

3. Type **0,'***SymbolFileName***'**, for example:

   0,'[Sys]<Vid>Vdm_Vga.sym'

4. Type **sbverrun=**

   The system responds *n:m*, where *n* is the DS of that partition, when you loaded the file at offset 0, for example, 30:0B630.

5. Subtract *n* from the original DS to find the offset, for example, 15D0-30 = 15A0.

[Print file]

Default: Screen only

If you want a record of command output, enter a file or printer specification. If you leave this field blank, command output appears on the screen only.

[Append to print file?]

Default: No

Enter **Yes** if you want to append command output to an existing file. If you enter **No** or leave this field blank, and the file already exists, the existing file is overwritten.

[Timer interval (50 ms)]

Default: 50 milliseconds

Specify the timer interval at which **Histogram** reads and reports results. Valid entries are in the range of 10 to 1000 milliseconds, in multiples of 10.

Operation

To use the **Histogram** command, follow these steps:

1. Execute the **Histogram** command. The following heading appears.

```
Time Running:___sec                    ACTION-SPACE start histogram
___% inside, ___% outside selected      TSS's FINISH exit histogram

TSS       CS:IP       count     %       Symbol  Histogrammer:  OFF
```

2. In a different partition, start the application on which you want **Histogram** to report.
3. To begin the histogram, press **ACTION-SPACEBAR**.

   The *Time running*, *% inside*, and *% outside* fields are filled in while the histogram is running.
4. Press **ACTION-BACKSPACE** to finish the histogram.

   The message "Sorting" appears while **Histogram** sorts the TSSs. Output appears on the screen and is optionally sent to the print file.
5. To exit, press **FINISH**.

Display

The histogram report contains the following columns of information:

TSS	Task segment selector
CS:IP	Code segment (CS) and instruction pointer (IP)
Count	Number of times this location was counted
%	Number of times the histogram recorded this location (given as a percentage of the total number of samples taken)
Symbol	Procedure name, if there is one, associated with this CS:IP

Function Keys

The function key menu is displayed at the bottom of the screen. Use these keys as follows:

F1 (Show All TSS)	Displays information about all samples taken, regardless of any *Options* form selections (the *Options* form is described below).
F2 (Show Sel TSS)	Displays information about selected TSSs only. (This option observes the TSS selections and the *CS Only* option.)
F3 (Show All Sym)	Displays information about all procedures. The symbol file must be loaded to use this option. With this option, **Histogram** looks at the CS:IP for each TSS and determines what symbol is associated with that address. Multiple occurrences of the same symbol within a given TSS are added together to yield the total time for that symbol. If a TSS has no symbol associated with it, it is probably in a system service or an operating system kernel, and the timing is given as in F1, above.

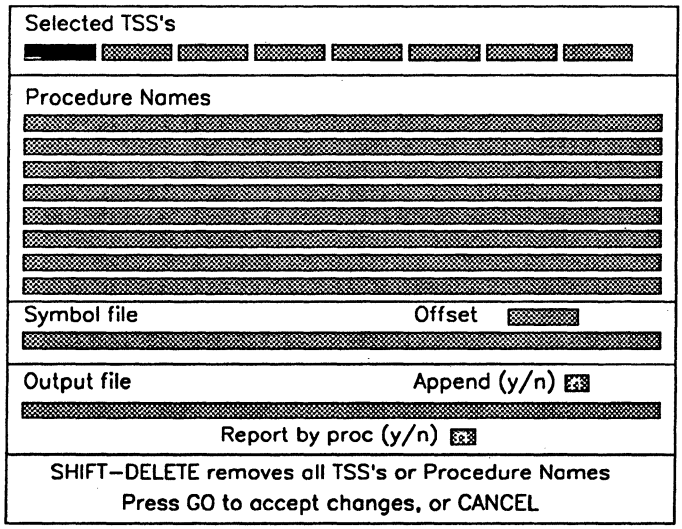
F4 (Show Sel Sym)	Displays information about selected procedures only. Procedures are specified through the *Options* form (described below).
F10 (Options)	Displays the *Options* form (described in detail below).

Options Form

When you press **F10**, the *Options* form, as shown in Figure 3-8, appears. To make entries in this form, press **RETURN** or **UP ARROW** to position the highlight, then type your selections.

When you have finished changing options, press **GO** to accept the changes or **CANCEL** to dismiss the form.

Figure 3-8. Histogram Options Form



Fields in the *Options* form are described below. Changes or additions you make in this form override command form entries.

Selected TSSs

Enter selected TSSs in these fields. Any TSSs you entered in the command form also appear here.

Procedure Names

Enter selected symbol (procedure) names. If you specify procedure names, you also must specify a symbol file (either here or in the command form). If you do not specify a symbol file, procedure names are ignored.

Symbol file

Enter a symbol file name if you want procedure names to be included in the histogram. If you specified a symbol file in the command form, it appears in this field.

Offset

For system services and operating system symbols, enter an offset (see the command form description, above).

Output file

See the *[Print file]* field, in the command form description.

Append

See the *[Append to print file?]* field in the command form description.

Report by proc?

By default, this function reports procedures by CS:IP, as shown in the following example:

```
0EC0 - FE6C:0057 00005
0EC0 - FE6C:0061 00003
0EC0 - FE6C:0064 00020
0EC0 - FE6C:006A 00002
0EC0 - FE6C:006D 00005
```

Specify **Yes** to display procedure names, in addition to CS:IP information, as shown in the following example.

```
0EC0 - FE6C:0057 00005    0.5 SYSTEM\InitMemory+3
0EC0 - FE6C:0061 00003    0.3 SYSTEM\InitMemory+0D
0EC0 - FE6C:0064 00020    2.2 SYSTEM\InitMemory+10
0EC0 - FE6C:006A 00002    0.2 SYSTEM\InitMemory+16
0EC0 - FE6C:006D 00005    0.5 SYSTEM\InitMemory+19
```

If a symbol file was specified, the symbol at CS:0 is displayed. In addition, multiple occurrences of the same symbol are compressed and reported as a single occurrence, as shown in the following example:

```
0000 - FE6C:0054 00035    4.0 SYSTEM\InitMemory
```

Install CDROM Service

Note: This command is not supported on EISA/ISA workstations.

Use this command to install the CD-ROM Service on a workstation. The CD-ROM Service is installed on the workstation to which the CD-ROM drive is connected. CD-ROM drives located on the server can be shared by all workstations in the cluster.

This command is implemented with the run file *CDROMService.run*.

Note: System services cannot be installed if Context Manager or the Remote User Manager is running on the processor. Therefore, they are frequently installed during system initialization, rather than with Executive commands. See "Installing System Services," in the CTOS System Administration Guide, *for more information.*

Command Form

```
Install CDROM Service
 [CDROM drive names]                                    _______
 [Maximum number of CDROM files open (300)]             _______
 [Maximum number of CDROM users (100)]                  _______
 [Name of CDROM Service (CDROM)]                        _______
```

Parameter Field

[CDROM drive names]

Default: See below

Unless you are using a customized version of the operating system, you can leave this field blank. By default, the CD-ROM Service recognizes devices named *zn*, where *n* is the sequential unit number of the nondisk SCSI device (for example, *z0*, *z1*, and so on).

If you are using a customized version of the operating system, enter the device name(s) defined for CD-ROM drives.

[Maximum number of CDROM files open (300)]

Default: 300

Specify the total number of CD-ROM files that can be open simultaneously by all users on all CD-ROM drives. For most purposes, the default of 300 is sufficient. For each file specified, 25 bytes of memory is allocated. Therefore, the default of 300 files requires 7,500 bytes. You can reduce memory requirements of the CD-ROM Service by specifying fewer files. The maximum value is 4095.

[Maximum number of CDROM users (100)]

Default: 100

Specify the total number of users who can access CD-ROM files simultaneously. For most purposes, the default of 100 is sufficient. For each user specified, 2 bytes of memory is allocated. You can reduce memory requirements by specifying fewer users. The maximum value is 65,535.

[CDROM device name (CDROM)]

Default: CDROM

Enter a device name for the CD-ROM drive. For this version, such a device name can be addressed by application programs only.

Install CFA File Filter

Use this command to install the Cluster File Access (CFA) File Filter. This allows the workstation to access disks on other workstations in the cluster that are running the CFA Workstation Agent.

Before workstations can access disks on other cluster workstations, the CFA Server Service must be installed on the server. In addition, the CFA Workstation Agent must be installed on those cluster workstations containing the disks that are to be accessed. See the **Install CFA Server Service** and **Install CFA Workstation Agent** commands, later in this section.

Install CFA File Filter has no command form. To execute it, type the command name, then press **GO**.

This command is implemented with the run file *CfaFF.run*.

Note: *System services cannot be installed if Context Manager or the Remote User Manager is running on the processor. Therefore, they are frequently installed during system initialization, rather than with Executive commands. See "Installing System Services," in the* CTOS System Administration Guide, *for more information.*

Install CFA Server Service

Note: *This command is supported on workstation servers only.*

Use this command to install the Cluster File Access (CFA) Server Service. It must be installed on the server before other workstations in the cluster can use the Cluster File Access facility, which allows users to access disks on workstations throughout the cluster.

See also the related commands, **Install CFA File Filter** and **Install CFA Workstation Agent**, for information about the CFA system services to install on cluster workstations.

Install CFA Server Service has no command form. To execute it, type the command name, then press GO.

This command is implemented with the run file *CfaM.run*.

Note: *System services cannot be installed if Context Manager or the Remote User Manager is running on the processor. Therefore, they are frequently installed during system initialization, rather than with Executive commands. See "Installing System Services," in the* CTOS System Administration Guide, *for more information.*

Install CFA Workstation Agent

Use this command to install the Cluster File Access (CFA) Workstation Agent. The CFA Workstation Agent allows other workstations in the cluster to access the local disks on the workstation where it is installed.

Availability and access levels of the disks are configured with the **CFA Configure** command, either before or after the CFA Workstation Agent is installed. See the documentation for that command, earlier in this section.

The CFA Server Service must be installed on the server before workstations can use the Cluster File Access facility. In addition, the CFA File Filter must be installed on workstations throughout the cluster that will be accessing disks on other cluster workstations. See the **Install CFA Server Service** and **Install CFA File Filter** commands, later in this section. See also the *CTOS System Administration Guide* for more detailed information about the Cluster File Access facility.

This command is implemented with the run file *CfaWA.run*.

Note: *System services cannot be installed if Context Manager or the Remote User Manager is running on the processor. Therefore, they are frequently installed during system initialization, rather than with Executive commands. See "Installing System Services," in the* CTOS System Administration Guide, *for more information.*

Command Form

```
Install CFA Workstation Agent
  [Config file ([d0]<Sys>CFAConfig.sys)]
  [Maximum remote clients (32)]
```

Parameter Field

[Config file ([d0]<Sys>CFAConfig.sys)]

Enter the file specification of the configuration file containing the disk access information you want to use. Access levels are assigned to disks with the **CFA Configure** command, described earlier in this section.

[Maximum remote clients]

Default: 32

Enter the maximum number of applications running elsewhere in the cluster that can simultaneously access disks on this workstation via the CFA Workstation Agent.

The maximum value that can be set is 256, however, depending on the configuration of the operating system, the full number specified in this field may not always be available to the CFA Workstation Agent.

Install Command Access Service

Use this command to install the Command Access Service, which limits access to the commands it monitors. The Command Access Service consists of a system service, a configuration file, and a log file, which are described below. See also "Limiting Access to Certain Commands," in the *CTOS System Administration Guide.*

This command is implemented with the run file *AccessService.run.*

Note: *System services cannot be installed if Context Manager or the Remote User Manager is running on the processor. Therefore, they are frequently installed during system initialization, rather than with Executive commands. See "Installing System Services," in the* CTOS System Administration Guide, *for more information.*

Command Form

```
Install Command Access Service
 [Maximum log file size (in sectors)] _____
 [Password]                           _____
```

Parameter Fields

[Maximum log file size (in sectors)]

Default: 10

Enter the number of sectors of disk space for the length of the log file. If you enter 0, logging is suppressed.

New entries are written at the beginning of the log file. As the log file becomes full, oldest entries are dropped from the end of the file.

[Password]

Default: No password

If the Command Access Service configuration file is password protected, enter its password here. See below for detailed information about the Command Access Service configuration file.

Configuration File

The Command Access Service reads a configuration file named *[!Sys]<Sys>UserCmdsConfig.sys*. It contains entries that permit or restrict access to the recognized commands. A sample file is pictured in Figure 3-9; keywords are described below:

Figure 3-9. Command Access Service Configuration File

```
:SignOnUserName: Tricia
:AllowedCommands: 'Cluster View'
                  'Set Time'
                  'Administrator Cluster View'
:RestrictedCommands:
:SignOnUserName: Jim
:AllowedCommands: 'Cluster View'
:RestrictedCommands:
:SignOnUserName: {Accts}Renee
:AllowedCommands: 'Cluster View'
:RestrictedCommands:
:SignOnUserName: Reed
```

:SignOnUserName:

This parameter defines a user name to the Command Access Service. To allow access to a user in another node, specify the node name before the user name, for example, *{Accts}Renee*.

:AllowedCommands:

This parameter is associated with a user name, as shown in Figure 3-9. It specifies commands to which the user has access. Valid values are **'Administrator Cluster View'**, **'Cluster View'**, and **'Set Time'** (including the single quotation marks).

:RestrictedCommands:

This parameter is associated with a user name, as shown in Figure 3-9. It specifies commands that are to be restricted by the Command Access Service. At this time, Standard Software commands do not make use of this feature.

Log File

When a user executes a monitored command, the Command Access Service writes an entry to the log file, *[!Sys]<Sys>Login.sys*. Each log-file entry contains the user's name, the date and time, and whether or not access was allowed, as shown in the following examples:

```
Renee  8/13/92 10:30 AM:  Cluster View - Access ALLOWED

Jim  8/13/92 10:30 AM:  Set Time - Access RESTRICTED
```

To obtain information about users who attempt to execute a monitored command, you must include their user names in the configuration file. For example, in Figure 3-9, Reed is not allowed access to any of the monitored commands. Therefore, if he does attempt to set the system date and time, the following entry will be written to the log file:

```
Reed  8/13/92 10:30 AM:  Set Time - Name only found
```

If, however, Reed's name is not listed in the configuration file, no log-file entry is made when he tries use one of the monitored commands.

Install DataComm Service

Note: *This command is not supported on EISA/ISA workstations.*

Use this command to install the DataComm Service on a workstation. It is required for the DCX Port Expander module.

Install the X-Bus Interface Service before you install the DataComm Service; see the **Install XBIF Service** command, later in this section.

Install DataComm Service has no command form. To execute it, type the command name, then press GO.

This command is implemented with the run file *DcxService.run.*

Note: *System services cannot be installed if Context Manager or the Remote User Manager is running on the processor. Therefore, they are frequently installed during system initialization, rather than with Executive commands. See "Installing System Services," in the* CTOS System Administration Guide, *for more information.*

Install Math Service

Use this command to install the Math Service. The Math Service provides floating point arithmetic calculations for workstations without a math coprocessor. In addition, it provides chip-level context management for real mode workstations with a math coprocessor. The Math Service is not required on protected mode workstations with a math coprocessor.

Math-coprocessor capabilities are required by programming languages that do not perform their own floating point arithmetic. See your programming language documentation to determine if you need to install the Math Service.

Install Math Service has no command form. To execute it, type the command name, then press GO.

This command is implemented with the run file *MathService.run*.

Note: *System services cannot be installed if Context Manager or the Remote User Manager is running on the processor. Therefore, they are frequently installed during system initialization, rather than with Executive commands. See "Installing System Services," in the* CTOS System Administration Guide, *for more information.*

Install MCR Service

Note: *This command is not supported on EISA/ISA workstations.*

Use this command to install the Magnetic Card Reader Service on a workstation. It allows users to sign on to the workstation by inserting an ID card into a magnetic card-reading device.

Install MCR Service has no command form. To execute it, type the command name, then press GO.

This command is implemented with the run file *MCRInstall.run*.

Note: *System services cannot be installed if Context Manager or the Remote User Manager is running on the processor. Therefore, they are frequently installed during system initialization, rather than with Executive commands. See "Installing System Services," in the* CTOS System Administration Guide, *for more information.*

Install Mouse Service

Use this command to install the Mouse Service on a workstation.

Install Mouse Service has no command form. To execute it, type the command name, then press GO. See also the related command, **Set Mouse Controls**, later in this section.

This command is implemented with the run file *Mouse.run*, command case 00.

Note: *System services cannot be installed if Context Manager or the Remote User Manager is running on the processor. Therefore, they are frequently installed during system initialization, rather than with Executive commands. See "Installing System Services," in the* CTOS System Administration Guide, *for more information.*

Install New Requests

Use the **Install New Requests** command to merge the contents of a binary request file with *[Sys]<Sys>Request.sys*, the binary request file that is loaded by the operating system when the system is bootstrapped.

Install New Requests performs the function of the **List Request Set** command by converting binary requests into a text file. It then performs the function of the **Make Request Set** command by merging those requests into *[Sys]<Sys>Request.sys*. See the **List Request Set** command and the **Make Request Set** command, in this section, for more information.

This command is implemented with the run file *SubCmd.run*, using the submit file *InstallNewRequests.sub*.

Command Form

```
Install New Requests
 Loadable request file  _____________________________
 [Destination path]     _____________________________
 [Exclusion list]       _____________________________
```

Parameter Field

Loadable request file

Enter the file name(s) of the binary loadable request file you want to merge with the system request file, *[Sys]<Sys>Request.sys*.

[Destination path]

Default: *[Sys]<Sys>*

Enter the volume and directory where *Request.sys* is located. This field is generally used to specify *[!Sys]<Sys>*, to merge requests on the server.

[Exclusion list]

Default: None

Enter the hexadecimal values of any request codes you want to exclude from the merge. You can use an at-file in this field.

Install Queue Manager

Use this command to install the Queue Manager on workstation servers and shared resource processors. On an SRP, **Install Queue Manager** is executed via Cluster View (see "Cluster View," in the *CTOS System Administration Guide*). The Queue Manager is used for scheduling spooled printing, Batch jobs, and other background processing.

When the Queue Manager is installed, it allocates memory as follows:

- For dynamic queue slots, which are later defined as queues, when print spoolers or the Batch Manager is installed
- For static queues, which are defined in *[Sys]<Sys>Queue.index*, when the system is booted

See the *Generic Print System Administration Guide* for more information about the Queue Manager.

This command is implemented with the run file *InstallQMgr.run*, command case 00.

Note: *System services cannot be installed if Context Manager or the Remote User Manager is running on the processor. Therefore, they are frequently installed during system initialization, rather than with Executive commands. See "Installing System Services," in the* CTOS System Administration Guide, *for more information.*

Command Form

```
Install Queue Manager
  [Use cache?]                                  _____
  [Maximum number of dynamic queues (0)]        _____
```

Parameter Fields

[Use cache?]

Default: Yes

If you enter **Yes**, or leave this field blank, the number of cache slots allocated is equivalent to the number of queues (up to a maximum of 126), as determined by the second parameter. Using the cache increases the speed of the Queue Manager because unmodified queue entries are kept in memory, thereby decreasing the number of times the disk is accessed.

If you enter **No**, no cache slots are allocated. In this case, the Queue Manager uses less memory but runs slower.

[Maximum number of dynamic queues (0)]

Default: See below

Enter the number of queues that you want to allocate in addition to those named in *[Sys]<Sys>Queue.index*. The total maximum number of queues is 255.

If you leave this field blank, space is allocated only for those queues named in *Queue.index*. However, if *Queue.index* is empty or does not exist, ten queues are allocated, to allow for installation of the Generic Print System.

The number of queue slots determines the size of the cache. Allocating more queues than you actually need will increase the size of the cache. Although this uses more memory, a larger cache may improve the Queue Manager's performance.

Install Screen Print

Note: *This command is not supported by all applications, for example, Presentation Manager or CTOS/Vpc™. See the SRA for Standard Software for more detailed information.*

Use this command to install the Screen Print Service. When it is installed, you can capture screen output in a file or send it to a printer. Screen output can be captured in ASCII, Generic Print Access Method (GPAM), or Tagged Image File (TIFF) formats. Those options are described in more detail below.

Note: *System services cannot be installed if Context Manager or the Remote User Manager is running on the processor. Therefore, they are frequently installed during system initialization, rather than with Executive commands. See "Installing System Services," in the* CTOS System Administration Guide, *for more information.*

This command is implemented with the run file *InstallScreenPrint.run*, command case 00.

Command Form

```
Install Screen Print
 Destination printer/file                    _____
 [Graphics action key (<G>)]                 _____
 [Text action key (<COPY>)]                  _____
 [Allow graphics output?]                    _____
 [Courier font size (10)]                    _____
 [Characters per inch (12)]                  _____
 [Left border (0.5 inch)]                    _____
 [Top border (1.0 inch)]                     _____
 [Framing?]                                  _____
 [Landscape printing?]                       _____
 [Single file for all copies?]               _____
 [Overwrite ok? - single file only]          _____
```

Parameter Fields

Destination printer or file

Enter a file or printer specification. Valid values are file specifications, GPS device names, spooler queues, or direct printer names.

If you specify a file, the suffix *.txt* (for text), *.tif* (for graphics), or *.gpam* for (GPAM) is appended to the file name, depending on the output format. In addition, the system date and time are inserted between the file name you specify and the suffix, unless you specify **Yes** in the *[Single file for all copies?]* field (described below).

Output formats are described in more detail below.

[Graphics Action-Key (<G>)]

Default: ACTION-G

Enter the key sequence to be used for capturing graphics output or GPAM output. If the specified device is not capable of printing graphics or certain GPAM elements, this key sequence produces the same results as the text action key sequence (described below).

If you specified **Yes** in the *[Allow graphics output?]* field (see below), this key sequence captures graphics output files in uncompressed tagged image file format (TIFF). TIFF files can be viewed with OFIS Imager, Image Designer™, and other applications that recognize TIFF files and typically require about 60 sectors of disk space. Note that graphics output is supported only on workstations equipped with a graphics controller.

If you do not enable graphics output (see *[Allow graphics output?]*, below), this key sequence captures output in GPAM format. GPAM output is neither ordinary text output, nor is it graphics TIFF output. It is a format that can be printed via the GPS Print Manager and closely resembles the image you captured on the screen.

In addition, GPAM output supports framing, margins, variable font sizes, and landscape printing. Fields for those features are described below.

Note that GPAM prints the standard system font (*t1Sys.font*). If you are using a customized font file, output results are unpredictable.

[Text Action-Key (<COPY>)]

Default: ACTION-COPY

Enter the key sequence to be used for capturing text output in an ordinary text format.

[Allow graphics output?]

Default: No

Enter **Yes** if you want to capture TIFF output when you press ACTION-G (or the graphics action key you specified above).

If you enter **No** or leave this field blank, the graphics action key captures output in GPAM format (see *[Graphics action key]*, above).

Graphics mode is supported only on workstations equipped with a graphics controller.

[Courier size (10)]

Default: 10 point

If output is to a GPS printer, you can specify a size in points for the courier font. If output is to a GPAM file, the default value of 10 points will be used. This field does not apply to text or graphics output.

[Characters per inch (12)]

Default: 12

If output is to a GPS printer, you can specify the number of characters to print per horizontal inch. If output is to a GPAM file, the default value of 12 characters will be used. This field does not apply to text or graphics output.

[Left border (0.5 inch)]

Default: 0.5 inch

If output is to a GPS printer, you can specify the number of inches for the left margin. If output is to a GPAM file, the default value of .5 inch will be used. This field does not apply to text or graphics output.

[Top border (1.0 inch)]

Default: 1.0 inch

If output is to a GPS printer, you can specify the number of inches for the top margin. If output is to a GPAM file, the default value of 1 inch will be used. This field does not apply to text or graphics output.

[Framing?]

Default: No

This field applies to GPAM output only. Enter **Yes** if you want a frame printed around the output.

[Landscape printing?]

Default: No

This field applies to GPAM output only. Enter **Yes** if you want output printed in landscape mode; that is, horizontally on the page, rather than vertically.

[Single file for all the copies?]

Default: No

Enter **Yes** if you want all output written to the same file specification. Output can be appended or it can overwrite existing file contents; see *[Overwrite ok? - single file only]*, below.

If you enter **No** or leave this field blank, a new file is created for each screen capture. The date and time are inserted in the file name to uniquely identify each file.

[Overwrite ok? - single file only]

Default: No

If you are capturing all output in a single file, enter **Yes** if you want each subsequent screen capture to overwrite the one before.

If you enter **No** or leave this field blank, each subsequent screen capture is appended to the single output file. This option applies to text captures only.

Capturing Screen Output

After installing the Screen Print Service, you capture screen output as follows:

- To capture text and graphics, press **ACTION-G** (or the specified key sequence).
- To capture text only, press **ACTION-COPY** (or the specified key sequence).

The system beeps twice at the beginning and twice at the end of the screen capture operation. Do not modify the screen while the capture operation is in progress.

Five beeps indicate an error. Use the **PLog** command (described later in this section) to check the system error log for error information.

Printing International Characters

The printable character set for the Screen Print Service is included in the binary message file named *[Sys]<Sys>ScreenPrintMsg.bin*. If you want printed output to contain international characters, you must customize the message file, as described in the following steps:

1. Generate an editable message text file with the **List Message File** file command.
2. Edit the character translation table contained in lines 100 to 355 of the message text file.

   Within the message text file, line :100: indicates the character to be printed in place of the standard character represented by hexadecimal code 0h, and so on through line :355: for hexadecimal code FFh. If no entry exists for a particular character, that character will not be translated. If the entire table is absent, a default translation scheme is employed. Minimal translation is achieved by having one entry in the table on line :100: to translate nulls to blank spaces.
3. Generate a new binary message file with the **Create Message File** command.

See the **List Message File** and **Create Message File** commands, in this section, and "Customizing Standard Software," in the *CTOS System Administration Guide*. See also the *CTOS Procedural Interface Reference Manual* for a table of the standard character set.

Install Sequential Access Service

Note: *This command is not supported on EISA/ISA workstations.*

Use this command to install the Sequential Access Service. The Sequential Access Service provides access to quarter-inch cartridge (QIC), digital data storage (DDS), and half-inch tape drives on workstations and SRPs. The Sequential Access Service supports up to eight tape drives.

For workstations, the Sequential Access Service is installed on each workstation to which a tape drive is connected. It is not installed on cluster workstations that are accessing a tape drive located on the server.

For SRPs, best performance is obtained when the Sequential Access Service is installed on the processor to which the tape drive is connected. However, for a SCSI QIC tape drive, the Sequential Access Service can be installed on any processor; for a non-SCSI QIC tape drive, it can be installed on any processor except a DP or an SP. For a half-inch tape drive, the Sequential Access Service must be installed on the DP or SP to which the tape drive is connected.

This command is implemented with the run file *InstallSeqService.run*, command case IN.

Note: *System services cannot be installed if Context Manager or the Remote User Manager is running on the processor. Therefore, they are frequently installed during system initialization, rather than with Executive commands. See "Installing System Services," in the* CTOS System Administration Guide, *for more information.*

Command Form

```
Install Sequential Access Service
 [Device(s) ([QIC])]                          _____
 [Buffer pool size in Kb (64)]                _____
 [QIC interface slot - XE/SRP only (77)]      _____
```

Parameter Fields

[Device(s) ([QIC])]

Default: *[QIC]*

On workstations, assign a device name for each tape drive. Default names are *QIC* for the first tape drive and *Seq0* for a second tape drive. On X-Bus workstations, devices are numbered sequentially along the X-Bus, starting from the right of the processor. On X-Bus+ workstations, device numbering begins from top to bottom in the base unit, and continues first in the SCSI expansion unit to the left of the base unit, and then to the right of the base unit. See the *SuperGen Series 5000 Hardware Installation and Owner's Guide* for more detailed information.

On SRPs, enter the device name assigned to each tape drive in the operating system configuration file; there is no default. See "Configuring Shared Resource Processor Operating Systems," in the *CTOS System Administration Guide*, for information about identifying sequential access devices.

[Buffer pool size in Kb (64)]

Default: 64

Enter the number K bytes for the buffer pool for the Sequential Access Service. This buffer pool is used by all tape drives on the system. On SRPs, the buffer pool consumes memory on the processor on which you install the Sequential Access Service.

[QIC interface slot - XE/SRP only (77)]

Default: 77h

This parameter applies to SRPs only. Enter the slot number containing the processor to which the QIC tape drive is connected.

Install Spooler

Note: *Do not use this command with the Generic Print System (GPS). GPS uses its own spooler, which is installed as part of the Print Service. See the* Generic Print System Administration Guide.

Use the **Install Spooler** command to install the pre-GPS printing spooler on a server, cluster, or standalone workstation. A spooler is installed on each processor to which a printer is attached. The Queue Manager must be installed before the spooler. See the appendix about pre-GPS printing in the *Generic Print System Administration Guide.*

This command is implemented with the run file *Spooler.run*, command case 00.

Note: *System services cannot be installed if Context Manager or the Remote User Manager is running on the processor. Therefore, they are frequently installed during system initialization, rather than with Executive commands. See "Installing System Services," in the* CTOS System Administration Guide, *for more information.*

Command Form

```
Install Spooler
  [Spooler configuration file]          _____________
  [Print first char if form feed?]      _____________
```

Parameter Fields

[Spooler configuration file]

Default: *[Sys]<Sys>SpoolerConfig.sys*

Enter the file specification for the spooler configuration file you want to use.

[Print first char if form feed?]

Default: No

If you enter **Yes**, and the first character of a file is a form feed, a form feed takes place before the file is printed. If you enter **No** or leave this field blank, the form-feed character is ignored.

Install Statistics Service

Note: *This command is supported on workstation processors only.*

Use this command to install the Performance Statistics Service. It maintains performance statistics for various resources, such as files, disks, and processors. It also makes such statistics available to the **STAT** command and other applications that request them. See the **STAT** command, later in this section. See also the *CTOS Programming Guide* for more information about the Performance Statistics Service.

This command is implemented with the run file *Statistics.run*.

Note: *System services cannot be installed if Context Manager or the Remote User Manager is running on the processor. Therefore, they are frequently installed during system initialization, rather than with Executive commands. See "Installing System Services," in the* CTOS System Administration Guide, *for more information.*

Command Form

```
Install Statistics Service
 [Heap size in kilobytes (10)]                _____
 [Log heap size in kilobytes (10)]            _____
 [Maximum users (4)]                          _____
 [Maximum sessions/user (4)]                  _____
 [Maximum blocks/session (10)]                _____
```

Parameter Fields

[Heap size in kilobytes (10)]

Default: 10

Enter the amount of memory in kilobytes to be allocated for a statistics session. The maximum is 55.

[Log heap size in kilobytes (10)]

Default: 10

Enter the amount of memory in kilobytes to be allocated for a log session. The maximum is 55.

[Maximum users (4)]

Default: 4

Enter the maximum number of users who can use the Performance Statistics Service at one time. The maximum is 10.

[Maximum sessions/user (4)]

Default: 4

Enter the maximum number of sessions a user can have open at one time. The maximum is 16.

[Maximum blocks/session (10)]

Default: 10

Enter the number of blocks that can be queried in a given session. This number is a combination of a block and an index in the Performance Statistics Structure. The maximum is 20.

Install Voice Service

Note: *This command is not supported on EISA/ISA workstations.*

Use this command to install the Voice Service. This command is for use on X-Bus+ workstations and on other workstations that include a Voice Processor™ or Telephone Manager module. It identifies the hardware configuration of the workstation and then installs the appropriate system service for voice and data capabilities.

Install Voice Service has no command form. To execute it, type the command name, then press **GO**.

This command is implemented with the run file *InstallVoice.run*, command case 00.

Note: *System services cannot be installed if Context Manager or the Remote User Manager is running on the processor. Therefore, they are frequently installed during system initialization, rather than with Executive commands. See "Installing System Services," in the* CTOS System Administration Guide, *for more information.*

Install XBIF Service

Note: *This command is not supported on EISA/ISA workstations.*

Use this command to install the X-Bus™ Interface Service, which is required to use certain workstation modules. It is for use on workstations only.

Install XBIF Service has no command form. To execute it, type the command name, then press **GO**.

This command is implemented with the run file *XBif.run* command case XS.

Note: *System services cannot be installed if Context Manager or the Remote User Manager is running on the processor. Therefore, they are frequently installed during system initialization, rather than with Executive commands. See "Installing System Services," in the* CTOS System Administration Guide, *for more information.*

Install XC002 Service

Note: *This command is not supported on EISA/ISA workstations.*

Use this command to install the XC-002 Service on a workstation. It is required for the XC-002 Port Expander module. Install the X-Bus Interface Service before you install the XC-002 Service; see the **Install XBIF Service** command, earlier in this section.

Install XC002 Service has no command form. To execute it, type the command name, then press GO.

This command is implemented with the run file *XC002Service.run*.

Note: *System services cannot be installed if Context Manager or the Remote User Manager is running on the processor. Therefore, they are frequently installed during system initialization, rather than with Executive commands. See "Installing System Services," in the* CTOS System Administration Guide, *for more information.*

Installation Manager

Use the **Installation Manager** command to install applications from floppy diskettes or tape distribution media, or from public software on the server. **Installation Manager** maintains a database of installed applications so that you can remove applications and perform software installations from the server.

See also the related commands, **Floppy Install**, **Tape Install**, and **Server Install**, which start the **Installation Manager** for a specific installation medium.

This command is implemented with the run file *InstallMgr.run*, command case IM.

Note: *The installation instructions for certain older applications instruct you to use the* **Install, Software Installation,** *or* **XESoftware Installation** *command. Those commands are now obsolete and are replaced by the* **Installaton Manager** *command. Features of the* **Installation Manager**, *however, such as public installation and installing from the server, are not available to such applications.*

Command Form

```
Installation Manager
  [Non-interactive?]
  [Append to log file?]
```

Parameter Field

[Non-interactive?]

Default: No

This field does not apply to floppy installation media. Enter **Yes** if you are installing from tape or from the server and want to install software in noninteractive mode. In noninteractive mode, the installation script does not pause for user input.

If you enter **No** or leave this field blank, the installation script may pause so that you can select options or enter other input, depending on the product.

[Append to log file?]

Default: No

Enter **Yes** if you want to append to, rather than overwrite, an existing log file. If you enter **No** or leave this field blank, the installation log file is overwritten with each successive installation procedure.

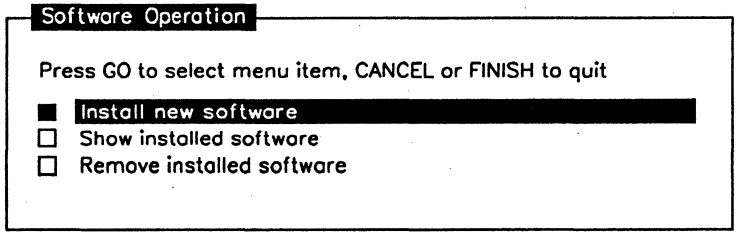
Operation

The **Installation Manager** command displays a series of menus from which you make selections. To select menu options, you can use the keyboard or a mouse, as described below:

- To use the keyboard, position the highlight with the arrow keys, then press GO.
- To use a mouse, position the highlight with the mouse, then press the **Mark** button. (See "Mouse," in Section 2, for more detailed information.)

After starting the **Installation Manager**, the Software Operation menu appears, as shown in Figure 3-10.

Figure 3-10. Installation Manager Software Operation Menu



541.3–10

Software operations are briefly described below. Each operation is described in detail later in this section.

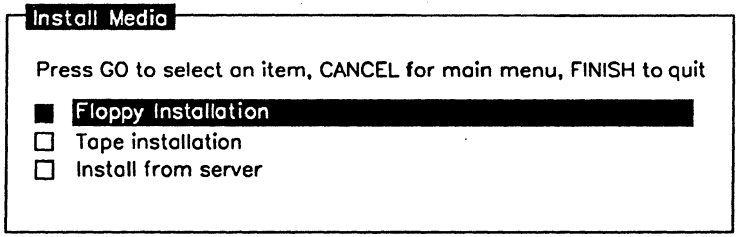
Install new software	Select this option to install or update software applications.
Show installed software	Select this option to display a list of software applications that have been installed by the **Installation Manager**.
Remove software	Select this option to remove files, commands, and request codes associated with a software application installed by the **Installation Manager**.

If you select an operation and then change your mind, press CANCEL to return to the Software Operation menu.

Installing Software

When you select *Install new software*, the Install Media menu appears, as shown in Figure 3-11.

Figure 3-11. Installation Manager Install Media Menu



541.3-11

Installation Media

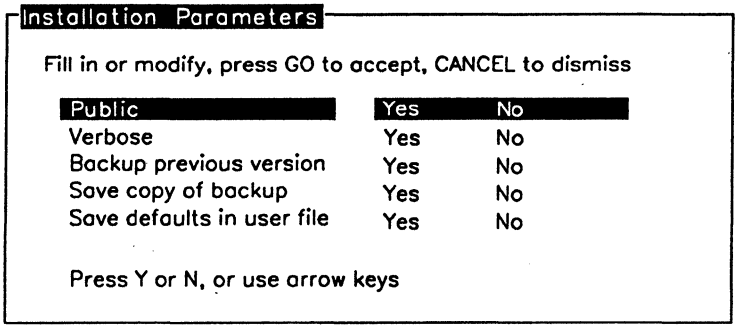
The installation media are described below:

Floppy installation	Select this option to install software from floppy diskettes.
Tape installation	Select this option to install software from tape. You can use either a tape drive located on your local workstation or on the server.
Install from server	Select this option to install software from the server onto a cluster workstation. If you select this option, the screen displays a list of software on the server that is available to the **Installation Manager**.

After selecting an installation media, choose *Continue Installation* or *Examine / Change Defaults*. When you choose *Examine / Change Defaults*, the Installation Parameters menu is displayed, as shown in Figure 3-12. Default values are highlighted. Default values are specified in the user configuration file; see "Customizing User Environments," in the *CTOS System Administration Guide*.

To change a value, press **Y** for Yes, **N** for No, or click your selection with the mouse. Installation parameters are described below.

Figure 3-12. Installation Manager Parameters Menu



541.3–12

Installation Parameters

Public

Select **No** for a private installation, which installs software on the local workstation.

Select **Yes**, for a public installation, which installs software on the server. Publicly installed software can be used by all workstations on the cluster. Note that for an application to be installable on cluster workstations from the server, it must be installed as public software on the server.

Verbose

Select **Yes** for a verbose installation. During a verbose installation, all messages and command output from the installation script are displayed on the screen.

Select **No** for a silent installation. During a silent installation, only a few progress messages are displayed.

Backup previous version

Select **Yes** if you want to back up the current version of a software product before the new version is installed. Unless specified otherwise in the *Archive path* field, which appears later if you select this option, backups are written to *[Sys]<Installed>*, for private installations, or *[!Sys]<Installed>*, for public installations, with the file prefix *Private* or *Public*, respectively.

Select **No** if you do not want to back up the current version.

Save copy of backup

Select **Yes** if you want to save a copy of the backup after the installation has been successfully completed. You can use this backup later to restore the previous version if necessary, for example, if you discover that the newer version is not compatible with your hardware or software configuration.

Select **No** if you do not want to save a copy of the backup.

Save defaults in user file

Select **Yes** to change your user file entries to the parameters you just selected. They become the new defaults for the Installation Manager.

Select **No** if you do not want to change your user file.

Installation Manager Prompts

Depending on the parameters you specify, you may be prompted interactively for the following information:

Archive path

If you entered **Yes** in *Backup previous version?*, this field appears. It is the volume and directory to which the backup will be written. Defaults are *[Sys]<Installed>* for private installation and *[!Sys]<Installed>* for public installations.

Software destination

This field shows the volume where the application will be installed. Change it if you want to install on a different volume. Defaults are *[Sys]* for local installations and *[!Sys]* for public installations.

Password

If a password is required to access the software destination directory, you are prompted to enter it, and the installation will not continue until a valid password is supplied.

Note, however, that only a single password can be active on a system at any given time. Therefore, if you are installing to a destination volume that requires a different password from *[Sys]<Sys>*, the installation may fail. For example, if **PswdA** is required to access the software destination volume but **PswdB** is required to access *[Sys]<Sys>*, the Installation Manager may not be able to create new commands in *[Sys]<Sys>*. Similarly, if the destination volume does not require a password, you will not be prompted to enter one; however, if a password is required for *[Sys]<Sys>*, the installation may fail.

To avoid such a problem, use the **Set Directory Protection** command to remove the directory password from *[Sys]<Sys>* before you begin the installation. Then, after the installation is complete, reassign the original password to *[Sys]<Sys>*.

Tape spec

Enter the tape specification for the application you want to install. In many cases, the default of *[QIC]0* is correct. The only way to be sure of this, however, is to look it up in the installation guide or Software Release Announcement for the product.

User name

This field shows the user name you are currently signed on with. During software installation, this is frequently not your usual user name. Enter your usual user name if you want to update the installation database and save defaults for it, rather than for the user name appearing in this form.

CmConfig file

If Context Manger is installed during the installation, the name of the active CM configuration file appears in this field. If Context Manager is not installed the default is *[Sys]<Sys>CmConfig.sys*. If you want the application added to Context Manager, enter the file specification of the CM configuration file you usually use.

Command file

If you are performing a private installation, the name of the active Executive command file appears in this field. If you want new commands placed in a different command file, enter its file specification. During public installations, new commands are always placed in *[!Sys]<Sys>Cluster.cmds* and this field is skipped.

Listing Installed Software

When you select *Show installed software* from the Software Operations menu, you are prompted to select either *Private software* or *Public software.*

To display the list of software applications for the current user name, select *Private software.* To display a list of software applications installed on the server, which may be shared throughout the cluster, select *Public software.*

Removing Installed Software

When you select *Remove installed software* from the Software Operations menu, you are prompted to select either *Private software* or *Public software.* A list of applications for the current user name, along with their version numbers, is displayed.

To remove an application, position the highlight, then press **GO**. The **Installation Manager** deletes run files, commands, requests, and configuration files associated with that application.

Recovering From Installation Failures

If you specified **Yes** to *Backup previous version?*, you can restore the previous version if the installation fails. To do so, press **GO** when the following message is displayed:

```
Press GO to restore the backup, CANCEL to finish.
```

To determine what caused the failure, see *[Sys]<Installed>Install.log*, the installation log file. See also the release documentation for the application and the *CTOS System Administration Guide.*

Restarting Installations

If an installation fails, you might be able to restart the installation at the point from which it failed. If an application supports the restart feature, and if it failed with a recoverable error, the following message is displayed when you restart the installation:

```
The previous installation of Product Name failed.

Press GO to restart installation, CANCEL to start new
installation.
```

Be sure to correct the problem that caused the initial failure before you restart an installation procedure. See the release documentation for the application for detailed installation instructions.

LCopy

Use the **LCopy** command to copy disk files only. In contrast, the **Copy** command can be used for copying to and from other devices, such as a printer or the keyboard; see "Device Specifications," in Section 2.

LCopy provides more sophisticated handling of parameter values than **Copy**, resulting in a powerful command for dealing with groups of files. With **LCopy**, you can divide a file specification into three parts:

- A prefix, which can include the volume and directory, as well as a file name prefix
- A file name
- A suffix

LCopy combines the prefix, file name, and suffix to form a file specification. If any of these parts is not specified, the value null string is used instead (null string means it contains no characters).

If you want to copy files to a new volume or a new directory without renaming them, you can do this more easily with **LCopy** than with **Copy**. You can also change the name of a file or group of files by adding a prefix or suffix with the **LCopy** command. **LCopy** is also useful for copying all the files needed for an application program onto a floppy disk, using an at-file specification.

If **LCopy** fails within a file list, press F1 to retry copying the current file. This feature is handy, for example, if you are copying to a floppy and forget to put the diskette into the drive, or if the diskette becomes full before you have copied all the files.

To specify passwords for the source and destination files, use the circumflex (^), as described in "Parameter Fields," below. See also "File Specifications," in the *CTOS Executive User's Guide*.

This command is implemented with the run file *LCopy.run*.

Command Form

```
LCopy
 File list                    _____________________________
 [File prefix(s) from]        _____________________________
 [File prefix(s) to]          _____________________________
 [File suffix(s)]             _____________________________
 [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 an at-file or wild card characters in this field.

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. It is necessary to specify these parts of the file specification here (rather than in *[File prefix(s) from]*), if the parameter value contains expandable characters, such as wild cards or an at-file.

[File prefix(s) from]

Default: Null string

Enter the string of characters you want prefixed to the file name in the *File list* field to construct the source file specification. If the number of entries in this field is equal to the number of entries in the *[File prefix(s) to]* field, entries are paired on a one-to-one basis.

If only one source file is specified (using any combination of *File list* and *[File prefix(s) from]*, the *[File prefix(s) to]* field can contain multiple prefixes; see *[File prefix(s) to]*, below.

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 in this field apply to the source file(s) if no password is specified in the *File list* field.

[File prefix(s) to]

Default: Null string

Enter the string of characters you want prefixed to the file name in the *File list* field to construct the destination file specification. If the number of entries in this field is equal to the number of entries in the *[File prefix(s) from]* field, entries are paired on a one-to-one basis.

If only one source file is specified (using any combination of *File list* and *[File prefix(s) from]*), the *[File prefix(s) to]* field can contain multiple prefixes; **LCopy** then makes a copy to each prefix specified in this field.

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 in this field apply to the destination file(s) if no password is specified in the *[File suffix(s)]* field.

[File suffix(s)]

Default: Null string

Enter the string of characters you want to use 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 in this field apply to the destination file(s) if no password is specified in the *[File prefix(s) to]* field.

[Overwrite ok?]

Default: Ask for confirmation

If you enter **Yes** and the destination file already exists, the existing file is automatically overwritten. If you enter **No** and the destination file already exists, the file is not copied. If you leave this field blank, you are prompted for confirmation before **LCopy** overwrites an existing file.

[Confirm each?]

Default: No

If you enter **Yes**, you are prompted to confirm the copy operation for each file. If you enter **No** or leave this field blank, 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**, the command terminates and returns to the Executive when an error is encountered. If you leave the field blank, **LCopy** stops and you are prompted for confirmation to continue.

[Verify copy?]

Default: No verification

If you enter **Yes**, the contents of the source and destination files are compared. If you enter **No** or leave this field blank, no comparison is made between the source and destination files.

Examples

The following example copies a group of files from the default volume and directory to a floppy diskette:

```
LCopy
 File list                Payroll.*
 [File prefix(s) from]
 [File prefix(s) to]      [f0]<Accounting>
 [File suffix(s)]
 [Overwrite ok?]
 [Confirm each?]
 [Continue on error?]
 [Verify copy?]
```

The following example copies all files in the *<Unisys>* directory of a floppy diskette to *[Sys]<Sys>*. Note that the volume and directory are specified in File list rather than *[File prefix(s) from]* to accommodate wild card expansion outside of the default path.

```
LCopy
 File list                [f0]<Unisys>*
 [File prefix(s) from]
 [File prefix(s) to]      [Sys]<Sys>
 [File suffix(s)]
 [Overwrite ok?]          Yes
 [Confirm each?]
 [Continue on error?]
 [Verify copy?]           Yes
```

The following example copies a single file to two destination directories.

```
LCopy
 File list                Abc.run
 [File prefix(s) from]    [f0]<Unisys>
 [File prefix(s) to]      [Sys]<Sys> [!Sys]<Sys>
 [File suffix(s)]
 [Overwrite ok?]          Yes
 [Confirm each?]
 [Continue on error?]
 [Verify copy?]
```

Librarian

Use the **Librarian** command to create and maintain libraries of object modules. See the **Librarian** command in the *CTOS Programming Utilities Reference Manual: Building Applications*, for detailed information about the command form and parameter fields.

Command Form

```
Librarian
 Library file                  _____________________________
 [Files to add]                _____________________________
 [Modules to delete]           _____________________________
 [Modules to extract]          _____________________________
 [Cross-reference file]        _____________________________
 [Suppress confirmation?]      _____________________________
 [Library block size]          _____________________________
 [Case sensitive?]             _____________________________
```

Link

Use the **Link** command to combine object modules into a version-4 run file. See the **Link** command in the *CTOS Programming Utilities Reference Manual: Building Applications*, for detailed information about the command form and parameter fields.

Command Form

```
Link
 Object modules
 Run file
 [Map file]
 [Publics?]
 [Line numbers?]
 [Stack size]
 [Max memory array size]
 [Min memory array size]
 [Runfile mode]
 [Version]
 [Libraries]
 [DS allocation?]
 [Symbol file]
 [Copyright notice?]
 [File to append]
```

Link V6

Use the **Link V6** command to combine object modules into a version-6 run file. See the **Link V6** command in the *CTOS Programming Utilities Reference Manual: Building Applications*, for detailed information about the command form and parameter fields.

Command Form

```
Link V6
 Object modules
 Run file
 [Map file]
 [Publics?]
 [Line numbers?]
 [Stack size]
 [Max array, data]
 [Min array, data]
 [Run file mode]
 [Version]
 [Libraries]
 [DS allocation?]
 [Symbol file]
 [Copyright notice?]
 [File to append]
 [Source debugger]
 [Linker config file]
```

Link V8

Use the **Link V8** command to create version-8 run files for use with a virtual memory operating system. See the **Link V8** command in the *CTOS Programming Utilities Reference Manual: Building Applications*, for detailed information about the command form and parameter fields.

Command Form

```
Link V8
 Object modules
 Run file
 [Map file]
 [Publics?]
 [Line numbers?]
 [Stack, Dgroup heap size]
 [Max array, data]
 [Min array, data]
 [Run file mode]
 [Version]
 [Libraries]
 [DS allocation?]
 [Symbol file]
 [Copyright notice?]
 [File to append]
 [Source debugger]
 [Linker config file]
```

List

Use the **List** command to display an unsorted list of file names. This command is internal to the Executive and displays the file list quickly, without chaining to *Files.run*.

This command is implemented with the Executive internal value !8.

Command Form

```
List
  [File list (*)]
```

Parameter Fields

[File list (*)]

Default: All files in default directory

Enter the file specifications of the files you want to list.

List Message File

Use the **List Message File** command to create a message text file from a binary message file. This command is useful when you do not have access to the original text file.

This command is implemented with the run file *ListMsgFile.run*.

Command Form

```
List Message File
 Binary file  _____________________________________
 [Text file]  _____________________________________
```

Parameter Fields

Binary file

Enter the file specification of the binary message file.

[Text file]

Default: See below

Enter a file specification for the message text file. The default is the name of the binary file, with the suffix *.txt* instead of *.bin*.

List Request Set

Use the **List Request Set** command to translate loadable binary request files into text-file output. It merges all requests in the files you specify into a single output file. That output file is then used with the **Make Request Set** command to create a single loadable request file from the merged command output. See the **Make Request Set** command, later in this section.

See the *CTOS Operating System Concepts Manual* and the *CTOS System Administration Guide* for more information about loadable request files.

This command is implemented with the run file *ListRqSet.run*, command case 01.

Command Form

```
List Request Set
 Request.n.sys file(s)          _____________________
 [Output file]                  _____________________
 [Include padded requests?]     _____________________
 [Exclusion list]               _____________________
```

Parameter Fields

Request.n.sys file(s)

Enter the name(s) of the binary request file(s) to be translated into text. If you specify more than one file, listed output is merged into a single text file. See the *[Output file]* field, below.

[Output file]

Default: Screen

Enter a file specification to contain the text output. If you leave this field blank, output appears on the screen.

[Include padded requests?]

Default: No

If you enter **Yes**, all requests from the closest lower 1000h will be listed. For example, if you have a loadable request 6010h, you will get padded requests for 6000h through 600Fh. See the operating system Software Release Announcement for more information.

If you enter **No** or leave this field blank, requests are not padded. With protected mode operating systems, it is not necessary to pad numerical sequences in a loadable request file.

[Exclusion list]

Default: None

Enter the hexadecimal values of any request codes you want to exclude from the listed request set.

Lock In Cache

Use the **Lock In Cache** command to lock a file or files into the file system cache. The file system cache is an area of memory where disk sectors are stored dynamically, as they are used. When sectors are cached, they are accessed from memory, rather than from disk. This increases the speed at which files are retrieved.

In general, only a few files should be locked into the cache, since this decreases the space available for dynamic caching. Applications that are integral to system performance can be locked into the cache.

For more information, see "Using a File System Cache," in the *CTOS System Administration Guide*. See also the related commands, **Disable Caching**, **Enable Caching**, **Unlock Cache**, and **Cache Status**, in this section.

This command is implemented with the run file *CacheUtility.run*, command case 03.

Command Form

```
Lock In Cache
 File list      _____________________________________
 [Print file]   _____________________________________
```

Parameter Fields

File list

Enter the name of the file(s) to lock into the cache. You can use wild cards or an at-file in this field.

[Print file]

Default: Screen only

If you want a record of screen output, enter a file or printer specification. If the file you specify does not exist, it is created. If it already exists, its contents are overwritten.

If you leave this field blank, output is sent to the screen only.

Logout

Use the **Logout** command to end a user session and return to the SignOn screen.

If you are using Context Manager and other applications are running, the **Logout** command does not execute. Instead, the Executive application exits and the following message is displayed on the Context Manager screen:

```
Warning: There are active contexts. Press GO to log out or CANCEL to deny.
```

To avoid losing data, properly exit all applications before logging out.

Logout has no command form. To execute it, type the command name, then press **GO**.

This command is implemented with the run file *SignOn.run*.

Maintain File

Use the **Maintain File** command to perform the following functions on Standard Access Method (STAM) files:

- Verify file structures
- Remove malformed records
- Remove deleted records
- Capture bad records in a log file

This command can be executed on the following types of STAM files:

- Record Sequential Access Method (RSAM) files
- Direct Access Method (DAM) files
- Data-store files of Indexed Sequential Access Method (ISAM) data sets

See the *CTOS Operating System Concepts Manual* and the *CTOS Procedural Interface Reference Manual* for information about RSAM, ISAM, and DAM files. See also the *ISAM Manual* for information about ISAM files.

This command is implemented with the run file *MaintainFile.run*.

Command Form

```
Maintain File
  Input files                  _____________________________
  [Output file]                _____________________________
  [Print file]                 _____________________________
  [Remove deleted records?]    _____________________________
  [Suppress confirmation?]     _____________________________
  [Capture bad data?]          _____________________________
```

Parameter Fields

Input files

Enter a list of one or more STAM file names. If this is the only field you fill in, the file structure is verified, and a report appears on the screen.

[Output file]

Default: No new file created

Enter the name of a file to which valid input records are copied. If the file already exists, its contents are overwritten. If it does not exist, it is created.

If all the input files are readable using DAM, the output file is a DAM file; otherwise, the output file is an RSAM file.

See the *[Remove deleted records?]* and *[Capture bad data?]* fields for information about the treatment of deleted and malformed records in the output file.

[Print file]

Default: Screen only

If you want a log file, enter a file or printer specification. The log file includes the number of records and bytes of data processed, the logical file address, the length of malformed records, and if specified, a copy of the malformed records.

[Remove deleted records?]

Default: 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 enter **No**, 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 records are created in the output file.

[Suppress confirmation?]

Default: No

If you enter **No** or leave this field blank, you are asked for confirmation before the command continues processing when a malformed record is encountered.

If you enter **Yes**, the command proceeds without confirmation; malformed records are handled as described under *[Remove deleted records?]*, above.

[Capture bad data?]

Default: No

Enter **Yes** if you want a copy of malformed records to be written to the log file specified in the *[Print file]* field.

Example

The form below merges two ISAM data sets, *<A>Ord.ISAM* and *<B>Ord.ISAM*, into the ISAM data set *<Srvr>Temp*; it also creates a log file for the verification report.

```
Maintain File
  Input files                  <A>Ord.ISAM <B>Ord.ISAM
  [Output file]                <Srvr>Temp
  [Print file]                 <Srvr>Temp.log
  [Remove deleted records?]
  [Suppress confirmation?]
  [Capture bad data?]
```

Maintain File scans the data-store files of the two input files, verifying file structures and reclaiming space occupied by deleted records; it then merges the valid records into *<Srvr>Temp*.

Make NLS

Use the **Make NLS** command to create a new version of the nationalization file, *[Sys]<Sys>Nls.sys*.

This command is implemented with the run file *SubCmd.run*, using the submit file *MakeNls.sub*.

Command Form

```
Make NLS
  [Version]
```

Parameter Fields

[Version]

Default: None

Enter a version identification to be assigned to the nationalization file. You can enter letters, numerals, and punctuation. If you use spaces, surround the entire entry with a pair of single quotation marks (').

Operation

To make changes to *Nls.sys*, edit the source file, *Nls.asm*. Then, execute **Make NLS**, which assembles and links *Nls.asm* to create *Nls.sys*.

After executing **Make NLS**, reboot the system to implement changes in the new version of *Nls.sys*.

Make Request Set

Use **Make Request Set** to create binary request files or to add new requests to existing binary request files. It is used most frequently by programmers and occasionally by system administrators.

Make Request Set converts the text form of a request file into a loadable binary request file. If more than one text file is specified, **Make Request Set** merges the binary output into a single request file. It also checks for and reports duplicate request numbers and other errors. Use the **List Request Set** command, described earlier in this section, to translate existing binary request files into text files before merging them with this command.

See the *CTOS Operating System Concepts Manual* and the *CTOS System Administration Guide* for more information about loadable request files.

This command is implemented with the run file *MakeRqSet.run*.

Note: *In most cases, it is not necessary to use this command after installing or updating a software application. When required by an application,* **Make Request Set** *is executed as part of the software installation procedure.*

Command Form

```
Make Request Set
 Text file(s)            _____________________________
 Binary request file     _____________________________
 [Error file]            _____________________________
 [Version]               _____________________________
```

Parameter Fields

Text file(s)

Enter the name(s) of the text file(s) containing the requests you want to merge into a loadable request file. (See the **List Request Set** command, earlier in this section, for information about converting existing binary request files into text files.)

Binary request file

Enter the file specification of the binary request file into which requests will be merged. If the file does not exist, it will be created.

[Error file]

Default: See below

Enter a file specification for the error log file. If the command fails, errors will be written to this file.

If you leave this field blank, the suffix *.lst* is added to the name of the binary request file to create the error log file.

You can also enter a device specification in this field. For example, to display errors on the screen, enter **[VID]**.

[Version]

Default: Date and time

The loadable request file can be given a version number for identification purposes (for example to match the version number of the system service that uses it).

Enter a string of letters, numbers, and punctuation, for example, *1.0-Test*. If you use spaces, surround the entire entry with a pair of single quotation marks ('). Without the single quotes, only characters up to the first space are included in the version number.

Make Translation File

Use the **Make Translation File** command to create a file that translates individual characters into other characters or strings before sending output to the printer. This command allows escape sequences to be sent to printers and also enables printing of characters that are mapped to a nonstandard ASCII character set.

See the **Create Configuration File** command, earlier in this section, for information about implementing this file on a specific printer channel.

This command is implemented with the run file *MakeTxlFile.run*.

Command Form

```
Make Translation File
  Source file name                  _____________________________
  Translation file name             _____________________________
```

Parameter Fields

Source file name

Enter the name of the text file that defines character translations. See "Source File," below, for information about creating one.

Translation file name

Enter a file specification for the translation file. You then specify this file when using the **Create Configuration File** command to create a printer configuration file.

Source File

The translation source file contains hexadecimal codes to represent characters or escape sequences. Translation file entries are written in the following format:

XX=YY

where

XX is the hexadecimal code for the actual character, and

YY is the hexadecimal code for the character or escape sequence you want to print.

The following example shows an entry that translates the pound sign (#) to a printer escape sequence:

23=1D

See the documentation for your printer for information about printer escape sequences.

To substitute a string of characters for a single character, entries are written in the following format:

XX = YY1, YY2,...YYn

where

YY1 through *YYn* are the hexadecimal codes for the translated string.

Md

The **Md** command is identical to the **Create Directory** command. Use it to create a new directory on a CTOS volume, and to establish directory passwords and protection levels.

Md is provided for users who are already familiar with DOS or UNIX. See the **Create Directory** command, earlier in this section, for a detailed description of the command form and parameter fields.

Merge

Use the **Merge** command to merge files of sorted data records according to embedded sort keys. See the *CTOS Sort / Merge Programming Reference Manual* for a detailed description of the command form, parameter fields, and operating instructions.

Command Form

```
Merge
 Input files                 _____________________________
 Output file                 _____________________________
 Keys                        _____________________________
 [Print file]                _____________________________
 [Suppress confirmation?]    _____________________________
```

Merge Command Files

Use the **Merge Command Files** command to merge two command files into one. It is most commonly used to add or update commands during software installation.

Command files contain command names and parameter fields for Executive commands. For detailed information, see "Command Files," in Section 2. For information about creating and removing commands, see the **Command File Editor, New Command,** and **Remove Command** commands, in this section.

This command is implemented with the run file *CmdFileEditor.run* using command case MC.

Command Form

```
Merge Command Files
 Command file from  _____________________________
 Command file to    _____________________________
 [Overwrite ok?]    _____________________________
 [Default volume]   _____________________________
```

Parameter Fields

Command file from

Enter the file specification of the command file you want to merge. The contents of this file will be merged with the file you specify in the *Command file to* field (described below).

Command file to

Enter the file specification for the destination command file. The file you specify here will contain the output of the merge operation.

[Overwrite ok?]

Default: Ask for confirmation

Enter **Yes** to overwrite existing commands in the destination file. Enter **No** to prevent existing commands from being overwritten. If you leave this field blank, you are prompted for confirmation before existing commands are overwritten.

[Default volume]

Default: No change

Enter the volume name, including brackets, to be used in the run file specifications for the commands you are merging. For example, if you are merging the **Files** command with the run file specification *[Sys]<Sys>Files.run*, you can change it to *[!Sys]<Sys>Files.run* in the destination command file by specifying *[!Sys]* in this field.

Merge Message Files

Use the **Merge Message Files** command to create a translated message file for a current release from a translated message file for a previous release. For example, suppose you have already translated the message file *ExecMsg.txt* into French for a specified version of Standard Software. You can then use **Merge Message Files** to generate a new French version of *ExecMsg.txt* for the next version of Standard Software, as described below. For more information, see "Message Files," in the *CTOS System Administration Guide.*

This command is implemented with the run file *MergeMessageFiles.run.*

Command Form

```
Merge Message Files
 Previous US version      _____________________________
 New US version           _____________________________
 Previous translation     _____________________________
 New translation          _____________________________
```

Parameter Fields

Previous US version

Enter the file specification of the previous US version of the message file, for example, *12.1>ExecMsg.txt.*

New US version

Enter the file specification of the new US version of the message file, for example, *ExecMsg.txt.*

Previous translation

Enter the file specification of the previous translation of the message file, for example, *Fr>12.1>ExecMsg.txt.*

New translation

Enter the file specification of the new translation of the message file, for example, *Fr>ExecMsg.txt.* This is the file to be created by **Merge Message Files.**

Example

Suppose you have translated version *1.0>Msg.txt* into French and you want to merge it with version *2.0>Msg.txt*.

To merge the old and new files, **Merge Message Files** also requires both the previous US version and the new US version of the message file to be present (in this example, 1.0 and 2.0 English versions of *ExampleMsg.txt*). Fill in the command form as shown in the following example:

```
Merge Message Files
 Previous US version      1.0>Msg.txt
 New US version           2.0>Msg.txt
 Previous translation     Fr>1.0>Msg.txt
 New translation          Fr>2.0>Msg.txt
```

The newly created output file, *Fr>2.0>Msg.txt*, is a valid message file containing all messages for the new version. Although you will eventually translate new messages into French, the mixed-language file can be used in the meantime.

In the example shown above, **Merge Message Files** creates a new file named *Fr>2.0>Msg.txt*. It contains French language messages for messages common to both *1.0>Msg.txt* and *2.0>Msg.txt*. For new messages, the English string is copied into the French message file and flagged with the following comment line:

```
;???
```

For messages common to both versions, but that contain English wording changes, the US 2.0 version is used for the actual message, along with comment lines for the US and French 1.0 versions.

For example, the beginning of the US version of *1.0>Msg.txt* might look like this:

```
:1: " : End of file, EOF - Error 1"
:2: " : End of medium, EOM - Error 2"
:3: " : Inconsistency - Error 3"
:4: " : Operator intervention - Error 4"
:5: " : Syntax error - Error 5"
:7: " : Not implemented - Error7"
```

The translation for *Fr>1.0>Msg.txt* would be as follows:

```
:1: " : Fin de fichier rencontrée - Erreur 1"
:2: " : Fin de support rencontrée - Erreur 2"
:3: " : Incohérence - Erreur 3"
:4: " : Intervention opérateur - Erreur 4"
:5: " : Erreur de syntaxe  - Erreur 5"
:7: " : Non exécuté - Erreur 7"
```

The *2.0>Msg.txt* file might include the following changes:

```
:1: " : End of file, EOF - Error 1"
:2: " : End of medium (EOM) - Error 2"
:3: " : Inconsistency - Error 3"
:5: " : Syntax error - Error 5"
:6: " : Server workstation unavailable - Error 6"
:7: " : Not implemented - Error 7"
```

The merged message file (*Fr>2.0>Msg.txt*) would then look like this:

```
:1: " : Fin de fichier rencontrée - Erreur 1"
;???
:2: " : End of medium (EOM) - Error 2"
;1.0>Msg.txt:2: " : End of medium, EOM - Error 2"
;Fr>1.0>Msg.txt:2: " : Fin de support rencontrée -
Erreur 2"
:3: " : Incohérence - Erreur 3"
:5: " : Erreur de syntaxe  - Erreur 5"
;???
:6: " : Server workstation unavailable - Error 6"
:7: " : Non exécuté - Erreur 7"
```

Note that message 4 was deleted from the old *1.0>Msg.txt* and is omitted from the new French translation. New messages, such as message 6, are flagged with commented question marks (;???). For changed messages, such as message 2, the new US message is included for the actual message, and both the former US and French versions are included as comments.

MkDir

The **MkDir** command is identical to the **Create Directory** command. Use it to create a new directory on a CTOS volume, and to establish directory passwords and protection levels.

MkDir is provided for users who are already familiar with DOS or UNIX. See the **Create Directory** command, earlier in this section, for a detailed description of the command form and parameter fields.

Module Definition

Use the **Module Definition** command to create object modules that contain special data required by the Linker to build dynamic link libraries (DLLs).

The **Module Definition** command form is pictured below. See the *CTOS Programming Utilities Reference: Building Applications*, for a description of parameter fields and command operation.

Command Form

```
Module Definition
  Input file                _____________________________
  [Object module]           _____________________________
  [Import library]          _____________________________
  [List file]               _____________________________
  [Suppress warnings?]      _____________________________
  [Suppress ordinals?]      _____________________________
```

Move

Use the **Move** command to move files from one file specification to another. It can be used to move files to different volumes or directories or to change the names of files.

The **Move** command works similarly to the **LCopy** command, and the parameter fields are identical. In contrast to **LCopy**, however, when a file is moved, it no longer exists in its original location. See the description of **LCopy**, earlier in this section, for information about the handling of file prefixes and suffixes.

This command is implemented with the run file *LCopy.run*, using command case MM.

Command Form

```
Move
 File list                    _____________________________
 [File prefix(s) from]        _____________________________
 [File prefix(s) to]          _____________________________
 [File suffix(s)]             _____________________________
 [Overwrite ok?]              _____________________________
 [Confirm each?]              _____________________________
 [Continue on error?]         _____________________________
 [Verify copy?]               _____________________________
```

Parameter Fields

File list

Enter the name(s) of the file(s) you want to move. You can also use an at-file or wild card characters in this field.

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. It is necessary to specify these parts of the file specification here (rather than in *[File prefix(s) from]*), if the parameter value contains expandable characters, such as wild cards or an at-file.

[File prefix(s) from]

Default: Null string

Enter the string of characters you want prefixed to the file name in the File list field to construct the source file specification. The number of entries in this field must be equal to those in the *[File prefix(s) 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 in this field apply to the source file(s) if no password is specified in the *File list* field.

[File prefix(s) to]

Default: Null string

Enter the string of characters you want prefixed to the file name in the *File list* field to construct the destination file specification. The number of entries in this field must be equal to those in the *[File prefix(s) from]* 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 in this field apply to the destination file(s) if no password is specified in the *[File suffix(s)]* field.

[File suffix(s)]

Default: Null string

Enter the string of characters you want to use 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 in this field apply to the destination file(s) if no password is specified in the *[File prefix(s) to]* field.

[Overwrite ok?]

Default: Ask for confirmation

If you enter **Yes** and the destination file already exists, the existing file is automatically overwritten. If you enter **No** and the destination file already exists, the file is not copied. If you leave this field blank, you are prompted for confirmation before existing files are overwritten.

[Confirm each?]

Default: No

If you enter **Yes**, you are prompted to confirm the move operation for each file. If you enter **No** or leave this field blank, you are not prompted for confirmation.

[Continue on error?]

Default: Ask for confirmation

If you enter **Yes**, errors are reported and **Move** continues. If you enter **No**, the command terminates and returns to the Executive when an error is encountered. If you leave the field blank, **Move** stops and you are prompted for confirmation to continue.

[Verify copy?]

Default: No verification

If you enter **Yes**, the contents of the source and destination files are compared. If you enter **No** or leave this field blank, no comparison is made between the source and destination files.

MSDOS Read

Use the **MSDOS Read** command to copy one or more MS-DOS files from a pseudovolume (PSV) or DOS floppy diskette to a CTOS volume. (See the manual for your PC emulation software for information about PSVs.)

This command is implemented with the run file *MsReader.run*, command case 00.

Note: *High density DOS diskettes can be read only on workstations equipped with high-capacity floppy drives.*

Command Form

```
MSDOS Read
  [DOS file mask(s)]                              _____
  [From MSDOS drive or PSV, ([f0])]               _____
  [Drive password]                                _____
  [Map CR/LF to newline?]                         _____
  [Directory prefix from]                         _____
  [Include files from subdirectories?]            _____
  [File prefix to]                                _____
  [Overwrite ok?]                                 _____
  [Print file]                                    _____
```

Parameter Fields

[DOS file mask(s)]

Default: See below

Enter the existing MS-DOS file name(s). File names can contain up to eight characters and can include the wild-card characters (* and ?). If you do not specify a file name, you are queried about each file in the root directory of the MS-DOS drive you specify in the *[From MSDOS drive]* field.

[From MSDOS drive or PSV ([f0])]

Default: *[f0]*

Enter the name of the MS-DOS PSV or floppy drive.

[Drive password]

Default: None

Enter the password for the MS-DOS PSV.

[Map CR/LF to newline?]

Default: No change

Enter **Yes** to convert carriage return and line feed pairs (CR/LF) to CTOS line endings (a single LF character). If you enter **No** or leave this field blank, CR/LF pairs are not changed.

[Directory prefix from]

Default: Root directory

Enter the pathname for the file or files to be copied.

[Include files from subdirectories?]

Default: No

Enter **Yes** to copy files contained in any subdirectories of the pathname you specified. If you enter **No** or leave this field blank, files contained in subdirectories are not copied.

[File prefix to]

Default: Null string

Enter the string of characters you want prefixed to the file name(s) specified in the DOS file mask(s) field to construct the destination file specification.

[Overwrite ok?]

Default: Ask for confirmation

If you enter **Yes** and the destination file already exists, it is overwritten. If you enter **No** and the destination file already exists, no file is copied. If you leave the field blank, the Executive prompts you for confirmation before an existing file is overwritten. Press **GO** to confirm, **CANCEL** to deny, or **FINISH** to stop copying files.

[Print file]

Default: Screen only

If you want a record of screen output, enter a file or printer specification. If the file does not exist, it is created. If the file already exists, output is appended to its contents. If you leave this field blank, output is sent to the screen only.

MSDOS Write

Use the **MSDOS Write** command to copy a CTOS file to an MS-DOS pseudovolume (PSV) or DOS floppy diskette. (See the documentation for your PC emulation software for information about PSVs.)

This command is implemented with the run file *MsReader.run*, command case 01.

Note: *This command writes either regular-density or high-density diskettes, depending on the capability of the floppy drive.*

Command Form

```
MSDOS Write
 Filename(s)                                  _____________________
 [To MSDOS drive or PSV, ([f0])]              _____________________
 [Drive password]                             _____________________
 [Map newline to CR/LF?]                      _____________________
 [File prefix from]                           _____________________
 [Directory prefix to]                        _____________________
 [Overwrite ok?]                              _____________________
 [Print file]                                 _____________________
```

Parameter Fields

Filename(s)

Enter the name of the CTOS file(s) to be copied. Make sure the file name you specify is a valid MS-DOS file name; that is, it does not contain characters reserved for special use by MS-DOS. See your MS-DOS manuals for information about MS-DOS file name restrictions.

[To MSDOS drive or PSV ([f0])]

Default: *[f0]*

Enter the name of the MS-DOS PSV or floppy drive.

[Drive password]

Default: No password

Enter the password for the MS-DOS PSV.

[Map newline to CR/LF?]

Default: No change

Enter **Yes** to convert CTOS line endings to CR/LF pairs. If you enter **No** or leave this field blank, CTOS new lines are not changed.

[File prefix from]

Default: Null string

Enter the string of characters you want prefixed to the file name you specified in the *Filename(s)* field to construct the source file specification.

[Directory prefix to]

Default: Root directory

Enter the target pathname for the file or files to be copied.

[Overwrite ok?]

Default: Ask for confirmation

If you enter **Yes** and the destination file already exists, it is overwritten. If you enter **No** and the destination file already exists, no file is copied. If you leave the field blank, the Executive prompts you for confirmation before an existing file is overwritten. Press **GO** to confirm, **CANCEL** to deny, or **FINISH** to stop copying files.

[Print file]

Default: Screen only

If you want a record of screen output, enter a file or printer specification. If the file does not exist, it is created. If the file already exists, output is appended to its contents. If you leave this field blank, output is sent to the screen only.

New Command

Use **New Command** to create new Executive commands and their associated command forms. It is frequently used to add or update commands during software installations.

See also the **Command File Editor** command, earlier in this section, for additional information about adding and updating commands.

For detailed information about command files, see "Command Files," in Section 2. For information about parameter management, see the *CTOS Operating System Concepts Manual.*

This command is implemented with the run file *CmdFileEditor.run*, command case AC.

Command Form

```
New Command
 Command name      _____________________________
 Run file          _____________________________
 [Field names]     _____________________________
 [Description]     _____________________________
 [Overwrite ok?]   _____________________________
 [Case (00)]       _____________________________
 [Command file]    _____________________________
 [Public command?] _____________________________
```

Parameter Fields

Command name

Enter a name for the new command. This command name will 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 a pair of single quotation marks ('), as shown in the following example:

'Set Directory Protection'

Run file

Enter the file specification for the run file or the Executive internal value that will be executed by the command.

When creating a command for a submit file, the run file is either *SubCmd.run* or the Executive internal value *!21*. See the **Submit** command and the **Playback** command, later in this section.

[Field names]

Default: No parameter fields

Enter the parameter prompts for the new command form. Surround multiple word parameter prompts with a pair of single quotation marks ('); separate parameter prompts with a space, as shown in the following example:

```
New Command
 Command name          'Sample Command'
 Run file              [Sys]<Sys>Sample.run
 [Field names]         'Prompt 1' 'Prompt 2' 'Prompt 3'
 [Description]
 [Overwrite ok?]
 [Case (00)]
 [Command file]
 [Public command?]
```

If the parameter prompts do not fit in this field, create an at-file containing the parameter prompts you want to use. The following example shows parameter prompts as they would be contained within an at-file.

```
'Prompt 1'
'Prompt 2'
'Prompt 3'
'Prompt 4'
```

See "At-Files," in Section 2, for more information about substituting the contents of a file for a parameter value.

[Description]

Default: No description

Enter a brief description of the command to be displayed when the **HELP** key is pressed. Enclose multiple-word descriptions in single quotation marks (').

[Overwrite ok?]

Default: Ask for confirmation

If you enter **Yes** and the command name already exists, the new command overwrites the existing command. If you enter **No** and the command name already exists, the new command is not created. If you leave this field blank, you are prompted for confirmation before the existing command is overwritten.

[Case (00)]

Default: 00

Enter the two-character command case for the function that you want the command to invoke. Some run files are designed to invoke multiple functions. In such a case, this field is used to define the function of the run file to be invoked when the command is executed. For example, a single run file, *Mouse.run*, is used both to install and deinstall the Mouse Service. Command case 00 invokes the install function, while command case DM invokes the deinstall function.

Note that command cases are internal to the run file. When you create a command, you must know the two-character designator for the command case you want to use. Command case designators are described in the release documentation and manual for each product.

Note: *When using* **New Command** *in a submit file, surround the command case with a pair of small slashed zeros. This character is automatically inserted in submit files created with the* **Command File Editor**. *To enter it with the Editor application, insert the hexadecimal number* **A7**.

[Command file]

Default: Current command file

Enter the file specification for a command file, if you want the command to be created in a different command file than the one you are currently using.

An entry in this field is disregarded if you specify **Yes** in the *[Public command?]* field, which is described below.

[Public command?]

Default: No

If you enter **Yes**, the command is created in the public command file, *[!Sys]<Sys>Cluster.cmds*. Note that if you enter **Yes** here, any entry you may have made in the *[Command file]* field is ignored.

If you enter **No** or leave this field blank, the command is created as described in the *[Command file]* field above.

Wild Card Expansion

Most commands expand wild cards after **RETURN** is pressed (see "Wild Cards," in Section 2). However, you can control whether or not wild cards are expanded in command form fields.

To suppress expansion of wild cards, append an asterisk (*) to the parameter prompt, as shown in the following example:

```
New Command
  Command name          'Pretend Command'
  Run file              [Sys]<Sys>PretendCommand.run
  [Field names]         'File from*' 'File to*'
  [Description]         'Example only; not a command.'
  [Overwrite ok?]
  [Case (00)]
  [Command file]
  [Public command?]
```

Such an asterisk is not visible as part of the parameter prompt when the command form is displayed on the screen, so the resulting command form looks like this:

```
Pretend Command
  File from
  File to
```

The asterisk, however, remains known to the Executive and is expanded internally when you press **GO** to execute the command.

Password Fields

When a command form contains a password field, you can instruct the Executive to disguise the password when it is typed into the field. Each character of a disguised password appears as a pound sign (#).

To disguise a password, append a circumflex (^) to the parameter prompt, as shown in the following example:

```
New Command
 Command name          'Another Command'
 Run file              [Sys]<Sys>AnotherCommand.run
 [Field names]         'File list' '[Password]^'
 [Description]         'Example only; not a command.'
 [Overwrite ok?]
 [Case (00)]
 [Command file]
 [Public command?]
```

The circumflex is not visible as part of the parameter prompt when the command form is displayed on the screen. However, it remains known to the Executive, so the password is disguised, as shown in the following example:

```
Another Command
 File list    *.poem
 [Password]   #####
```

Partition Status

Use the **Partition Status** command to display information about memory partitions, the paging service, and dynamic link libraries (DLLs), including detailed information about individual partitions and user numbers, as well as a graphical representation of system memory usage.

For system administrators, the **Partition Status** command is helpful in configuring a system for optimal performance, as described in the *CTOS System Administration Guide*. To make use of certain statistics, however, it is necessary to understand topics such as user numbers, memory management, and dynamic link libraries, as described in the *CTOS Operating Systems Concepts Manual*.

This command is implemented with the run file *SystemMgr.run*, using command case PS.

Operation

Partition Status has no command form. To execute it, type the command name, then press **GO**. To exit **Partition Status**, press **FINISH**.

The **Partition Status** display flickers on the screen every 10 seconds as it is updated. To prevent updating and freeze the screen, press **CODE-F10**. To resume updating, press **CODE-F10** again. Updating also resumes when you select another display.

Main Display

After executing the **Partition Status** command, the main display appears. An example of the display as it appears on a protected-mode system without virtual memory is shown in Figure 3-13. On a real-mode system, the main display is identical to the one pictured in Figure 3-13; on a virtual memory system, the columns labeled *Allocated* and *Used* are labeled *Linear* and *Physical*, respectively.

If the entire display does not fit on the screen, press **SCROLL UP**.

Figure 3-13. Partition Status Main Display

```
Partition              Allocated      Used       Run file executing

   System                856K         856K       pClstrLfs  3.4
2  Vdm_VGA    3.4        184K         182K
3  Mouse Server           21K          19K
4  XvtChService          337K         335K
5  Primary               591K         230K       SystemMgr.run

Total RAM:              4096K                    Available:  2747K
```

Real Mode and Protected Mode Fields

The following fields are displayed for real-mode systems and protected-mode systems without virtual memory:

Partition — Contains the user number and name of the partition. The user number is assigned when the partition is created and is the same as the partition handle.

Allocated — Shows the size of the partition in K bytes, including overhead for the partition structures.

Used — Shows the amount of memory, in K bytes, currently being used in the allocated partition.

Run file executing — Shows the run-file name of the program currently executing. The field is blank if the program is a system service.

Total RAM — Displays the total amount of memory on the processor, before allocation.

Available — Displays the amount of memory that is not currently in use. Note, however, that this amount may include unusable fragments, which are not available as contiguous memory.

Virtual Memory Fields

The following fields are displayed for virtual memory systems:

Partition	Contains the user number and name of the program. Partitions are not created on virtual memory systems, so the user number becomes the means for keeping track of associated program processes.
Linear	Displays the amount of linear address space currently being used. Linear address space is virtual memory, up to four gigabytes, from which code and data can be accessed. It is possible for the value shown in this field to exceed the amount of actual physical memory.
Physical	Shows the amount of actual physical memory currently being used.
Run file executing	Shows the run-file name of the program currently executing. The field is blank if the program is a system service.
Totals	Displays the totals for the *Linear* and *Physical* columns, respectively.
Total DLL	Displays the total amount of memory consumed by DLLs.
Total RAM	Displays the total amount of physical memory on the processor, before allocation.
Available	Displays the amount of physical memory currently available on the processor.

Main Display Function Keys

While viewing the main display, the following function keys are available:

F1 (Details)	Displays detailed information about a specific partition. To select a partition, position the highlight, then press **F1**.
F2 (Map)	Displays a memory map or a frame usage chart.
F10 (Update)	Updates the display and unfreezes it if updating had been suspended.

Details Display

The Details screen displays detailed information about the partition or user number you select. It displays a basic set of fields for real-mode, protected-mode, and virtual memory operating systems, with additional fields being displayed for the paging service on virtual memory systems only. A different set of fields is also displayed for DLLs. All fields are described below.

Basic Fields

The following fields are displayed for run-files and system services on real-mode, protected-mode, and virtual memory operating systems:

Partition	Displays partition name listed on the main display.
Run file	Shows the name of the run file that is currently active.
Exit run file	Shows the name of the run file that will become active when the current run file exits.
User number	Shows the assigned user number.
Partition type	Shows the partition type.
Partition size	Shows the size of the partition or linear memory in K bytes.

Memory

Displays the amount of memory and starting address in linear memory for the following:

Long lived memory

Short lived memory

Code

Available (memory that is preallocated between long and short lived memory)

U structure (user structures)

LDT (local descriptor table)

Context status

Shows whether the run file is resident or swapped, and in the case of a system service, whether it is serving requests and whether swapping is disabled.

Sharing code

Indicates by Yes or No whether the active run file is sharing code with another partition. It also displays the number of partitions that are sharing the same code.

If the partition is sharing code, its size is displayed to include the shared code (see the description of the *Partition size* field, above). The actual size of a code-sharing partition can be obtained by subtracting the amount of code from the displayed partition size.

Resources used

Shows resources, such as the system timer and communications interrupt service routines used by the application.

Outstanding rqs

Shows the number of requests that are waiting to be served by the active run file.

Last erc

Shows the last status code returned by the operating system.

Mode

Displays the run-file mode.

Paging Service Fields

In addition to the fields described above, the following information about the paging service is displayed on virtual memory systems only:

Physical (Max, Cur, Lck)	Displays the maximum number of frames that should ever be allocated, the number of frames currently allocated, and the number of frames currently locked into memory. (A *frame* is 4K bytes of physical memory.)
Prefaulted pages	Displays the number of used and unused prefaulted pages. (A *prefaulted page* is paged into memory in anticipation of its being needed.)
Working set (Max, Cur)	Displays the maximum number of frames used and number of frames currently being used. (The *working set* is the number of frames the program needs to execute with reasonable forward progress.)
Waits for clean page	Displays the number of times the program had to wait for a page to be moved out of memory.
Total page faults	Displays the total number of page faults that have occurred since the application was started. (A *page* is 4K bytes of addressable code or data. A *page fault* occurs when a new page replaces the contents of a frame.)
In last 5 sec	Displays the number of page faults that have occurred in the last 5 seconds.
Total wait time	Displays the total time, in tenths of a second, spent waiting for page faults.
In last 5 sec	Displays the time, in tenths of a second, spent waiting for page faults during the last 5 seconds.

DLL Fields

If the user number for which you display details contains a DLL rather than a run file or system service, the following fields are displayed:

Library file	Displays the file specification of the DLL.
Library name	Displays the name of the DLL.
Memory	Displays the amount of memory and starting address in linear memory for the following fields: *Library code* *Library data* *Directory* *Resource*

At the bottom of the screen, statistics are displayed for the paging service; see "Paging Service Fields," earlier in this section, for a description of those fields.

Details Display Function Keys

While viewing the Details display, the following function keys are available:

F1 (Home)	Returns to the main display.
F2 (Map)	Displays a memory or frame distribution map.
F9 (Next)	Cycles through the Details display for each partition or user number.
F10 (Update)	Updates the display and unfreezes it if updating had been suspended.

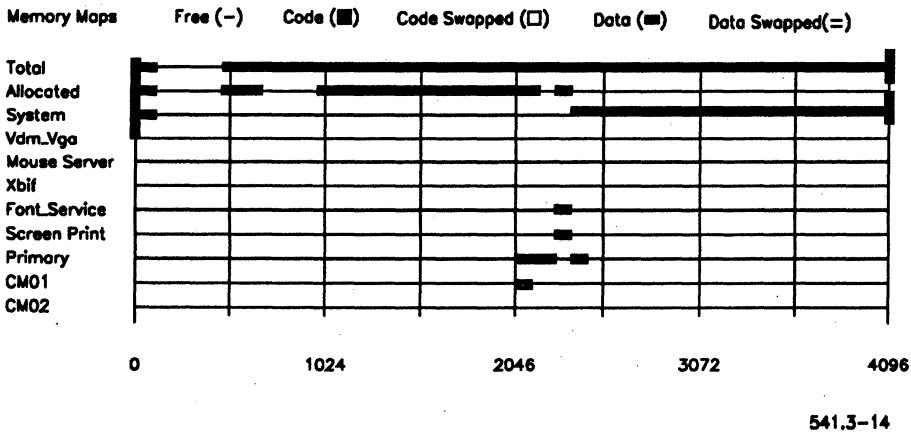
Map Display

The Map display shows a detailed memory map or a frame usage chart. Figure 3-14 pictures an example map for a protected-mode processor without virtual memory. The virtual-memory frame usage chart displays free and allocated frames only.

On the memory map, code is not distinguished from data for system-service run files, run files earlier than version 6, applications linked with data first, or applications containing noncontiguous code.

On real-mode processors, code and data are swapped back into their original locations as shown on the map display. On protected-mode processors without virtual memory, code and data are swapped back into any available location and swapping is depicted by a blank line on the map display.

Figure 3-14. Partition Status Map Display



Memory Map Usage Notation

On the Map display, memory usage is characterized as follows:

Ordinary line	Free memory or frames
Filled box	Code (a program) loaded into memory (real-mode operating systems and protected-mode without virtual memory only)
Outlined box	Code swapped out of memory (real-mode operating systems only)
Bold line	Data loaded into memory or allocated frames
Double line	Data swapped out of memory (real-mode operating systems only)

Map Display Function Keys

While viewing the Map display, the following function keys are available:

F1 (Home)	Returns to the main display.
F10 (Update)	Updates the display and unfreezes it if updating had been suspended.

Path

Use the **Path** command to set default values for node name, volume name, directory name, file prefix, and password. This information appears as the path setting in the status area of the screen, as shown in the following example:

```
Path: {Node}[Volume]<Directory>FilePrefix-
```

All parameter fields for this command are optional, so you only need to fill in the ones you want to change.

This command is implemented with the Executive internal value !7.

Command Form

```
Path
  [Volume]
  [Directory]
  [Default file prefix]
  [Password]
  [Node]
```

Parameter Fields

[Volume]

Default: No change

Enter a volume or device name. Square brackets ([ ]) are not required.

[Directory]

Default: No change

Enter a directory name. Angle brackets (< >) are not required.

[Default file prefix]

Default: No change

Enter a string of characters to be used as a file prefix. The file prefix, however, only applies to the default path, so it is not used if you override the path setting with a file specification.

To remove a file prefix from the path setting, enter a pair of single quotation marks (') in this field, as shown in the following example:

```
Path
  [Volume]
  [Directory]
  [Default file prefix]  ''
  [Password]
  [Node]
```

[Password]

Default: No change

Enter a volume, directory, or file password. The password entered here will be used for every command until you enter a new password, remove the password, or log out. The password does not appear in the path setting.

To remove a password from the path setting, enter a pair of single quotation marks (') in this field.

[Node]

Default: No change

Enter a node name; braces ({ }) are not required.

To remove a node name from the path setting, enter a pair of single quotation marks (') in this field.

Playback

Use the **Playback** command to execute simple submit files. In most cases, **Playback** is used to replay submit files that have been created with the **Record** command. It executes faster than the **Submit** command because it is internal to the Executive, however, it is not capable of expanding either system-defined or user-defined variables. See the **Submit** command, later in this section, for information about executing submit files that contain variables.

This command is implemented with the Executive internal value !21, command case 01.

Command Form

```
Playback
 File
```

Parameter Field

File

Enter the file specification of the submit file you want to execute.

PLog

Use the **PLog** command to display, and optionally print, the system log file. The system log file is named *[Sys]<Sys>Log.sys*; it is created when the volume is initialized.

System log file entries contain the date and time of the error, error type, and additional information about the error. The most recent errors appear at the beginning of the log file. As the log file fills, older entries are eliminated from the end of the file. Log buffer overflow (Error 290) indicates that multiple errors occurred rapidly and the operating system was unable to log them all.

See the *CTOS Status Codes Reference Manual* for a description of error codes.

This command is implemented with the run file *PLog.run*.

Command Form

```
PLog
  [Error type - B,Cl,Cr,D,In,Is,M (all)]  _____
  [Print file]                            _____
  [Volume name]                           _____
  [After date/time]                       _____
```

Parameter Fields

[Error type - B,Cl,Cr,D,In,Is,M (all)]

Default: All

Enter **B**, **Cl**, **Cr**, **D**, **In**, **Is**, or **M**, to indicate which type of errors you want displayed from the log file. If you leave this field blank, all error types are displayed. Individual error types are described below:

B	System bootstrap events (not necessarily errors)
Cl	Cluster communications errors
Cr	Crash
D	Disk errors
In	System initialization errors
Is	Indexed Sequential Access Method (ISAM) errors
M	Messages generated by applications

[Print to]

Default: Screen only

Enter a file or printer specification. Command output is always displayed on the screen.

[Volume name]

Default: *[Sys]*

Enter the volume name for the log file you want displayed. A log file is maintained only for system volumes and only if a log file length greater than zero was specified when the disk was initialized.

[After date/time]

Default: Current date and time

Enter the starting date and time from which you want to display error entries. If you enter a date only, errors are displayed from 12:00 midnight. Use a valid date/time format, as you would to sign on or set the system date and time (see **Set Time** command, later in this section).

Disk Errors

Status codes in the 300 range indicate disk errors. See the hardware manual for the disk for recommended solutions to the status codes you encounter.

If you note several errors for the same *Cylinder/Head/Sector* of a disk, or if it takes a number of times to recover data on a particular spot, enter this area in a bad spots file before you reinitialize the disk. See also **Format Disk**, earlier in this section, and "Correcting Input/Ouput (I/O) Errors," in the *CTOS System Administration Guide*, for more information about disk problems.

Examples

The following example shows a typical **PLog** entry for a system crash. See the *CTOS Status Codes Reference Manual* for a description of status code 26 and the recommended solution. Most users are not concerned with the *Description*, *Location*, and *Crash Information* fields. This information can be helpful, however, to system engineers or if you need to call Technical Support. Note that this entry also shows the workstation type, amount of memory, and type of operating system that was booted.

```
NGENT3,  Cluster WorkStation,  With File System
Memory size:  3584K,  SignOn User Name: Tricia
SYSTEM CRASH-(ERC=26)                    Fri Jun 10, 1988 12:17 PM
Description: Stray interrupt of type 11 detected
Executing instruction preceding location 0A03:0015
Crash information: 001AH 0047H 0047H 0000H 0000H 0089H 0A03H 0015H
Os Booted: pClstrLfs 3.4
```

The following example shows a **PLog** entry for a disk error. See the *CTOS Status Codes Reference Manual* for a description of status code 301; then see the disk hardware manual for a description of the *Command*, *Main Status*, and *Error Status* fields. Most users are not concerned with the information in those fields; however, it can be helpful to a field service engineer when repairs are needed.

```
NGENT3,  Cluster WorkStation,  With File System
Memory Size: 3584K,  SignOn User Name: Tricia
DISK ERROR - Floppy Unit 0 (ERC=301)   Thu Jun 2, 1988 12:30 PM
Description: Sector not found
Number of Retries: 8 (Unable to Recover),  Volume Name:
Cylinder: 0, Head: 0, Sector: 7, Number of Sectors: 1
Command: 88
Main Status: 10
```

PMake

Use the **PMake** command to convert a version 6 run file to a bootable version 4 run file.

When converting the run file, **PMake** uses the prototype descriptor information in the version 6 run-file header to construct the global descriptor table (GDT). **PMake** also appends a special code segment, *RealNub.sys*, to the system image, which contains code for entering protected mode.

This command is implemented with the run file *PMake.run*.

Command Form

```
PMake
 Run file                  _____________________________
 [Extra GDT slots (16)]    _____________________________
 [IDT slots (256)]         _____________________________
 [Load offset (0)]         _____________________________
 [Include symbols?]        _____________________________
 [Symbol file (-.sym)]     _____________________________
 [List file (-.gdt)]       _____________________________
 [Image file (-.img)]      _____________________________
 [Realnub file]            _____________________________
 [iSgFirst]                _____________________________
 [Compress?]               _____________________________
```

Parameter Fields

Run file

Enter the file name of the version 6 run file that is to be converted into a bootable protected-mode system image (for example, *pClstrLfs.run*).

[Extra GDT slots (16)]

Default: 16

Enter the number of GDT descriptors to be allocated, beyond what is indicated in the run-file header. **PMake** allocates a number of GDT descriptors based on the version 6 run-file header.

[IDT slots (256)]

Default: 256

Enter the number of interrupt descriptors that should be allocated. **PMake** allocates space in the system image for the interrupt descriptor table (IDT). The default value of 256 is the maximum number allowed.

Note that **PMake** initializes all IDT slots to 0.

[Load offset (0)]

Default: 0

Enter the load offset of the system image from the beginning of physical memory. The load offset must be in the range 0 to 65535.

[Include symbols?]

Default: No

Enter **Yes** to specify that the symbol file is to be included in the system image. System image resident symbols are used only by the diagnostic debugger.

[Symbol file (-.sym)]

Default: RunFileName.*sym*

Enter the file name of a symbol file that is to be included in the system image.

[List file (-.gdt)]

Default: RunFileName.*gdt*

Enter the file name of a report describing the system image.

[Image file (-.img)]

Default: RunFileName.*img*

Enter the file name of the output file, the bootable protected-mode system image.

[Realnub file]

Default: *[Sys]<Sys>Realnub.sys*

Enter the file name of the file that contains the code for entering protected mode.

The Realnub file is a special version 4 run file that contains the real-to-protected initialization. **PMake** sets the initial instruction pointer (CS:IP) of the bootable image to address this code. Following a bootstrap, Realnub executes to enter protected mode, initializes protected-mode registers, and jumps to the protected-mode starting address.

PMake places initial protected-mode register values used by Realnub in a communication area that immediately precedes Realnub.

[iSgFirst]

Default: Value specified in version 6 run-file header

Enter a value that specifies the index of the first nonreserved GDT slot. This slot is the first slot used to describe a segment.

[Compress?]

Default: No

Enter **Yes** to direct **PMake** to compress the image file by removing zeros from the run file. The zeros are recreated by the Realnub code. The Realnub code uses information in the communication area (the area that immediately precedes Realnub) to reconstruct the image.

Print

Use the **Print** command to add a file to the scheduling queue for spooled printing. Unlike the **Format** command, **Print** does not create a temporary file to be placed in the queue. Rather, the actual file is queued for spooled printing. Therefore, do not modify or delete the file until printing is complete.

For more information on the printing process, see the *Generic Print System Administration Guide*.

This command is implemented with the run file *Print.run*.

Command Form

```
Print
 File list                    _____________________________
 [Queue name (SPL)]           _____________________________
 [Number of copies]           _____________________________
 [Delete after printing?]     _____________________________
 [Special forms name]         _____________________________
 [Print wheel name]           _____________________________
 [Printing mode]              _____________________________
 [Align form?]                _____________________________
 [After date time]            _____________________________
 [Security mode?]             _____________________________
 [Priority]                   _____________________________
 [Confirm each?]              _____________________________
```

Parameter Fields

File list

Enter the name(s) of the file(s) to print.

[Queue name (SPL)]

Default: *[Spl]*

Enter the name of the scheduling queue to contain the queue entry generated by this print request. The name must match the queue name defined for the system.

[Number of copies]

Default: 1

Enter the number of copies you want to print.

[Delete after printing?]

Default: No

Enter **Yes** if you want the file to be deleted after it is printed.

[Special forms name]

Default: None

If you want to print the file on special paper or a preprinted form, enter a name for it; the 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 leave this field blank, the printer does not pause.

[Print wheel name]

Default: None

Enter the name of the print wheel to use; the name can have up to 12 characters. If a name is specified, the printer pauses before printing so the print wheel can be changed.

To restart the printer, invoke **Print Manager** for GPS printers or the **Spooler Status** command for non-GPS printers, and issue the appropriate command to restart the printer.

[Printing mode]

Default: Normal

Enter **ASCII** for normal mode, **WP** or **Plotter** for image mode, or **Binary** for binary mode. See the *Generic Print System Administration Guide* for more information on printing modes.

[Align form?]

Default: No

If you enter **Yes**, the printer pauses after the first page is printed, so that paper can be aligned. To restart the printer, invoke **Print Manager** for GPS printers or the **Spooler Status** command for non-GPS printers, and issue the appropriate command to restart the printer. If you enter **No** or leave the field blank, the printer does not pause for forms alignment.

[After date time]

Default: No delay

Enter the date and time after which you want the file to be printed. Use either of the following formats:

Apr 25, 1988 9:00 am

4/25/88 9:00 am

If you leave this field blank, the file is printed as soon as the specified printer is available.

[Security mode?]

Default: No

If you enter **Yes**, the printer pauses before the file is printed. You must then enter the password from the **Print Manager** or **Spooler Status** display on the workstation where the printer is connected.

[Priority]

Default: 5

Enter the priority number you want to assign to the print job. Priorities are 0 to 9, with 0 being the highest. Files with the highest priority print first.

[Confirm each?]

Default: No

If you enter **Yes**, you are prompted for confirmation before each file is printed.

Rd

The **Rd** command is identical to the **Remove Directory** command. Use it to remove a directory from a volume.

Rd is provided for users who are already familiar with DOS or UNIX. See the **Remove Directory** command, later in this section, for a description of the command form and parameter fields.

Read Hardware ID

Use the **Read Hardware ID** command to read a hardware identification number that has been assigned to a workstation with the **Write Hardware ID** command.

Read Hardware ID has no command form. To execute it, type the command name, then press **GO**.

This command is implemented with the run file *RHwId.run*.

Record

Use the **Record** command to record keystrokes in a file until the **Stop Record** command is executed. Recorded keystrokes are stored in a submit file, which can be replayed later with the **Submit** command.

The **Record** command can also be started with the **CODE-MARK** keystroke combination. This creates a submit file that can be played back later with a simple keystroke combination, rather than with a command. This feature is discussed under "Operation," below.

This command is implemented with the Executive internal value !19.

Note: *Commands recorded while the screen is set for narrow characters might not replay properly when the screen is set for large characters. To eliminate this problem, record* **Screen Setup** *to set the screen correctly, as the first command in your submit file.*

Command Form

```
Record
 File to record on _____________________________
```

Parameter Fields

File to record on

Enter a file specification for the submit file you are going to record.

Operation

After filling in the command form and pressing **GO**, the word "Recording" appears in the status area of the screen. While the Executive is recording, all keystrokes you issue are recorded in the submit file.

You can also initiate recording by pressing **CODE-MARK**. When you do so, the next keystroke you type identifies the character you will use to replay the submit file (described below). The "Recording" message then appears and subsequent keystrokes are recorded.

When you use the **CODE-MARK** method, the submit file is named as follows:

*[Scr]<Exec>*UserNameX.*sub*

where

UserName	Is the SignOn user name that appears in the status area of the screen.
X	Is one alphabetical or numeric character.

For example:

[Scr]<Exec>PattyA.sub

To replay a submit file recorded in this manner, simply press **CODE-*X***, then **GO**. (In the example shown above, this is **CODE-a**).

Note that these submit files are stored by user name. Therefore, several users who share a workstation can create unique submit files for their own user names.

Note: *On systems that use a memory disk for the* [Scr] *volume, the submit files for macros are stored in* [Sys]<Exec> *and then placed in* [Scr]<Exec> *on the memory disk as they are used. See the* CTOS System Administration Guide *for information about memory disks.*

Whether initiated with the **Record** command or with **CODE-MARK**, recording continues until you execute the **Stop Record** command or press **CODE-BOUND**.

Recorded submit files may be edited with a text editor, however, special keyboard codes are sometimes inserted into them as they are recorded. If you are editing a recorded submit file, do not change or remove character sequences that begin with ¢3 and end with ¢4 unless you fully understand 12-byte keyboard records, as described in the *CTOS Operating System Concepts Manual.*

Remove Command

Use **Remove Command** to delete a command name from an Executive command file. After a command name is removed, it is no longer recognized by the Executive.

This command is implemented with the run file *CmdFileEditor.run*, using command case RC.

Command Form

```
Remove Command
 Command list          _____________________________
 [Command file]        _____________________________
 [Public command?]     _____________________________
```

Parameter Fields

Command list

Enter the full command name of the command you want to remove. Abbreviations are not allowed in this field. If the command name contains spaces, enclose it in a pair of single quotation marks ('), as shown in the following example:

```
Remove Command
 Command list          'Unwanted Command'
 [Command file]        _____________________________
 [Public command?]     _____________________________
```

To remove more than one command, separate command names with a space, as shown in the following example, or use an at-file (see "At-Files," in Section 2).

```
Remove Command
 Command list          'This Command' 'That Command'
 [Command file]        _____________________________
 [Public command?]     _____________________________
```

[Command file]

Default: Currently active command file

If you want to remove a command in a command file other than the one you are currently using, enter the file specification for that command file.

An entry in this field is disregarded if you specify **Yes** in the *[Public command?]* field, which is described below.

[Public command?]

Default: No

If you enter **Yes**, the command is removed from the public command file, *[!Sys]<Sys>Cluster.cmds*. Note that if you enter **Yes** here, any entry you may have made in the *[Command list]* field is ignored.

If you enter **No** or leave this field blank, the command is removed as described in the *[Command list]* field above.

Remove Directory

Use the **Remove Directory** command to remove a directory from a volume. This command optionally deletes all the files in the directory before removing it.

This command is implemented with the Executive internal value !15.

> **Caution**
>
> This command can destroy data. If you choose to delete files in a directory before you remove it, those files are permanently removed from the disk and cannot be recovered.

Command Form

```
Remove Directory
 Directory name(s)                    _____________________
 [Volume or directory password]       _____________________
 [Delete all files in directory?]     _____________________
 [Confirm each while deleting?]       _____________________
```

Parameter Fields

Directory name(s)

Enter the name(s) of the directory you want to remove.

[Volume or directory password]

Default: Currently active password

Enter a valid volume or directory password.

[Delete all files in directory?]

Default: No

If you enter **Yes**, all files in the directory are deleted before the directory is removed. A directory cannot be removed until all files in it are deleted.

If you enter **No** or leave this field blank, and the directory contains files, the following message appears:

```
Directory is not empty, cannot remove it.
```

[Confirm each while deleting?]

Default: No

If you enter **Yes**, you are prompted to confirm the deletion of each file in the directory. If you enter **No** or leave the field blank, you are not prompted for confirmation; all files are deleted if you specified **Yes** in the *[Delete all files in directory?]* field (see above).

Rename

Use the **Rename** command to change the name of a file. You can also use it to move files from one directory to another on the same volume. (See the **Move** command, earlier in this section, for information about moving files from one volume to another.) Capitalization in file names can also be changed with the **Rename** command; for example, *myfile* can be changed to *MyFile*.

This command is implemented with the Executive internal value !4.

Command Form

```
Rename
 File from      _____________________________
 File to        _____________________________
 [Overwrite ok?] _____________________________
 [Confirm each?] _____________________________
```

Parameter Fields

File from

Enter the file name(s) of the file(s) you want to rename. You can use wild cards in this field.

File to

Enter a new file name for the file. You can use wild cards in this field, however, they must match, in number and pattern, those that you used in the *File from* field. If you want to move a file to a different directory, remember to include the directory name in this field.

[Overwrite ok?]

Default: Ask for confirmation

If you enter **Yes** and a file with the name you entered in the *File to* field already exists, the existing file is overwritten. If you enter **No** and a file with the name you entered in the *File to* field already exists, the existing file is not overwritten. If you leave the field blank, you are prompted for confirmation before existing files are overwritten.

[Confirm each?]

Default: No

If you enter **Yes**, you are prompted for confirmation before renaming each file.

Resource Librarian

Use the **Resource Librarian** command to create and maintain libraries of data resources within run files.

The **Resource Librarian** command form is pictured below. See the *CTOS Programming Utilities Reference: Building Applications*, for a description of parameter fields and command operation.

Command Form

```
Resource Librarian
 Input file (.run or .res)  _____________________________
 [Resources to add]         _____________________________
 [Resources to delete]      _____________________________
 [Resources to extract]     _____________________________
 [Resource config file]     _____________________________
 [Cross-reference file]     _____________________________
 [Suppress confirmation?]   _____________________________
```

Restore Archive

Use the **Restore Archive** command to restore files onto a volume from a backup created with either the **Volume Archive** or **Selective Archive** command.

You can use the **Restore Archive** command to do the following:

- Restore an entire volume or selected files, creating any needed directories on the destination volume
- Restore the characteristics of a file (creation date, protection level, and so forth) as they existed at the time of backup
- Optionally, list the files contained within the archive dataset, without restoring them
- Optionally, write a log file of the restore operation

The disk you restore files to must be a valid initialized volume.

Before you can restore data from a tape, the Sequential Access Service must be properly installed. See the **Install Sequential Access Service** command, earlier in this section, and "Installing System Services," in the *CTOS System Administration Guide.*

For information about performing backups, see the **Volume Archive** and **Selective Archive** commands, in this section, and "Backing Up and Restoring Data," in the *CTOS System Administration Guide.*

This command is implemented with the run file *RestoreArchive.run*, command case RA.

Command Form

```
Restore Archive
  [Archive dataset ([QIC])]          _____
  [File list from (<*>*)]            _____
  [File list to (<*>*)]              _____
  [Overwrite ok?]                    _____
  [Confirm each?]                    _____
  [Sequence number]                  _____
  [Merge with existing file?]        _____
  [List files only?]                 _____
  [Print file]                       _____
  [Suppress user interaction?]       _____
```

Parameter Fields

[Archive dataset ([QIC])

Default: *[QIC]*

If you are restoring from floppy diskettes or another hard disk, enter the file specification of the archive files. Exclude the two-digit sequence number, as shown in the following example:

[Volume]<Dir>d0Archive

If you are restoring from tape, enter the tape drive name and optionally the tape mark position of the backup, as shown in the following example:

[QIC]0

Tape drive names are defined when you install the Sequential Access Service; see the **Install Sequential Access Service** command, earlier in this section.

Tape mark specifications are as follows:

0	Specifies the beginning of the tape.
n	Is the number of a particular tape mark, such as 1, 2, and so on. The first dataset is 0, the second 1, the third 2, and so on.

If you do not specify a tape mark, the default is 0.

[File list from (<*>*)]

Default: *(*)[*]<*>**

Enter the list of files to restore. You can use wild cards or an at-file in this field. Valid file specifications take the following form:

(Node)[Volume]<Directory>FileName^password

Note that in most cases, particularly when restoring an archive dataset created with the **Volume Archive** command, you do not need to be particularly concerned with the full file specification. That is, to restore the complete archive dataset, you would simply specify **<*>***, since all files in the archive dataset would have been backed up from the same node and volume.

Theoretically, however, you can selectively back up and restore files from diverse nodes, volumes, and directories. See "Methods," below, for a detailed discussion of how file specifications and wild cards map from this field to the *[File list to]* field.

To restore all files in an archive dataset to their original directories on the default volume, you can leave this and *[File list to]* blank.

[File list to (<*>*)]

Default: Original directory on default volume

Enter the file specification to which files will be restored. They must map to the files specified in the *[File list from]* field. See "Methods," below, for a detailed discussion of how file specifications and wild cards map from the *[File list from]* field to this field.

[Overwrite ok?]

Default: Ask for confirmation

If you enter **Yes**, and a file on the volume to which you are restoring exists with the same name as a file from the archive data set, the disk file is overwritten with the file from the archive dataset. If you enter **No** the existing file is not overwritten. If you leave this field blank, you are prompted for confirmation before an existing file is overwritten.

Files containing volume structures, such as *<Sys>FileHeaders.sys* or *<Sys>Mfd.sys*, are not overwritten, since restoration of those files would destroy the volume structures on the valid volume to which you are restoring. If you need to restore such a file for trouble-shooting or debugging purposes, it can be restored from a selective backup to a file specification that is different from its original name.

[Confirm each?]

Default: No

If you enter **Yes**, you are prompted for confirmation before each file is restored. If you enter **No** or leave this field blank, files are restored without confirmation.

[Sequence number]

Default: 01

Enter the sequence number of the diskette or tape you want to start restoring from. For example, to start with the second diskette of an archive dataset, enter **02** as the sequence number.

[Merge with existing file?]

Default: No

If you enter **Yes**, and the archive file contains unreadable data (due to an input/output error occurring on the volume from which the archive dataset was obtained), any corresponding sectors in the destination file are bypassed and existing data is retained.

If you enter **No** or leave this field blank, and the archive dataset contains files that were backed up with unreadable sectors, corresponding sectors in the destination file specification are overwritten with 0s.

[List files only?]

Default: No

If you enter **Yes**, the files on the archive file are listed but not restored.

[Print file]

Default: Screen only

Enter a file or printer specification, if you want to write command output to a file or send it to a printer. If the file you specify does not exist, it is created. If it already exists, command output is appended to it. If you leave this field blank, output is sent to the screen only.

[Suppress user interaction?]

Default: No

Enter **Yes** to suppress messages that prompt for user interaction, such as mounting the archive medium. When user interaction is suppressed, **Restore Archive** exits with appropriate error messages. If you enter **No** or leave this field blank, **Restore Archive** pauses and prompts for correction of conditions that require user interaction.

Methods

The **Restore Archive** command can be used to restore almost any combination of files from an archive dataset. It is frequently used to restore a volume to its original state, by recreating directories and restoring files. It is also used to restore individual files, or to consolidate a number of files from diverse directories into a single directory.

The *[File list from]* and *[File list to]* fields may contain none, one, or many parameters. What you specify determines the end result of the restoration process. The following conditions apply to valid entries for these fields:

- If either node or volume is omitted from the file specification, defaults are *(*)* and *[*]* respectively.
- If *[File list to]* is not specified, it is considered to be *(*)[*]<*>** and is treated as described below for one parameter.
- If *[File list to]* has more than one parameter, *[File list from]* must have exactly the same number of parameters, each of which must map to its corresponding parameter in the *[File list to]* field (see "Rules for Using Wild-Card Characters," below).
- If *[File list to]* has only one parameter, *[File list from]* may have any number of parameters, each of which must map to the *[File list to]* parameter.

Rules for Restoring With Wild-Card Characters

The **Volume Archive** and **Selective Archive** commands preserve full file specifications, including node, volume, directory, and file name, in the archive dataset; therefore, it is possible to restore with many combinations of wild-card characters. If you specify invalid wild card combinations in the *[File list from]* and *[File list to]* fields, you are informed that the wild card patterns do not match.

Special rules for using wild-card characters with **Restore Archive** are described below. See "Wild-Card Characters," in Section 2, for general information.

- In node, volume, and directory names, wild-card characters can map on a one-to-one or many-to-one basis. The following example shows one-to-one mapping for node names, and many-to-one mapping for volume and directory names, where all volume names in the archive dataset map to *[Sys]* and all directory names map to *<Test>*:

```
[File list from (<*>*)]  {DV}[*]<*>*
[File list to (<*>*)]    {OS}[Sys]<Test>*
```

- In file names, the number and order of wild-card characters must be the same in both the *[File list from]* and *[File list to]* fields. The following example shows all files in the archive dataset with the suffix *.run* being restored to *<TestDir>* on the default volume. Note that the file name portion of this example follows the usual rules of wild card matching as described in Section 2:

```
[File list from (<*>*)]  <*>*.run
[File list to (<*>*)]    <TestDir>12.1-*.run
```

Restoring All Files to Their Original Directories

To restore all files from the archive dataset to their original directories on the current default volume, you can leave both the *[File list from]* and *[File list to]* fields blank. That is the same as specifying the following:

```
[File list from (<*>*)]  <*>*
[File list to (<*>*)]    <*>*
```

To restore all files from the archive dataset to their original directories on a local volume other than the default, specify the following:

```
[File list from (<*>*)]  <*>*
[File list to (<*>*)]    [VolName]<*>*
```

To restore all files from the archive dataset to their original directories on another node, specify the following:

```
[File list from (<*>*)]  <*>*
[File list to (<*>*)]    {NodeName}[VolName]<*>*
```

Restoring All Files to a Single Directory

You can restore all files from their original directories in the archive dataset to a single directory on the specified volume. The following example shows how to restore all files in an archive dataset to *[Sys]<Sys>*:

```
[File list from (<*>*)]  <*>*
[File list to (<*>*)]    [Sys]<Sys>*
```

Restoration of Passwords

When an archive dataset is created from an unencrypted volume, directory and file passwords are stored in the archive dataset and preserved when it is restored. If a directory or file has no password when it is backed up, the active path password or, if present, the password supplied in the *[File list to]* field is applied to it during restoration.

When a archive dataset is created from an encrypted volume, no passwords are backed up. During restoration, encrypted passwords would become reencrypted and, therefore, would be meaningless on the receiving volume.

Resume Cluster

Use the **Resume Cluster** command to restart cluster operations after they have been disabled with the **Disable Cluster** command. Files that were open on cluster workstations before cluster communications were disabled, however, are not automatically reopened. For an operation to resume without user intervention, an application program must reopen the files.

Resume Cluster has no command form. To execute it, type the command name, then press **GO**.

This command is implemented with the run file *Cluster.run*, using the command case 00.

RmDir

The **RmDir** command is identical to the **Remove Directory** command. Use it to remove a directory from a volume.

RmDir is provided for users who are already familiar with DOS or UNIX. See the **Remove Directory** command, earlier in this section, for a description of the command form and parameter fields.

Run

Use the **Run** command to execute a run file when a command has not been created for it. When you execute the **Run** command with parameters, a variable length parameter block (VLBP) is built. This is the same result as when parameters are supplied through an Executive command form.

The **Run** command can also be used to run programs with the Debugger. If you press **CODE-GO** (instead of **GO**) after filling in the **Run** command form, the program is loaded but not started. Control is then passed to the Debugger.

This command is implemented with the Executive internal value !18.

Command Form

```
Run
 Run file
 [Case]
 [Command]
 [Parameter 1]
 [Parameter 2]
 [Parameter 3]
 [Parameter 4]
 [Parameter 5]
 [Parameter 6]
 [Parameter 7]
 [Parameter 8]
 [Parameter 9]
 [Parameter 10]
 [Parameter 11]
 [Parameter 12]
 [Parameter 13]
 [Parameter 14]
 [Parameter 15]
 [Parameter 16]
```

Parameter Fields

Run file

Enter the file specification of the run file you want to execute, for example, *[Sys]<Sys>Files.run*. A run file consists of executable code that has been assembled and linked.

[Case]

Default: 00

Enter a two-character case value. A case value is intrinsic to the run file. Case values are used to make programs execute in different ways. For example, the run file *SeqAccessUtility.run* has multiple case values: case ER invokes the tape erase function, while case RT invokes the tape retension function. For more information about command case, see the *CTOS Operating System Concepts Manual*.

[Command]

Default: No name

Enter the command name you want to use to execute the run file.

[Parameter 1]

Default: None

Enter the parameter value(s) you would include in the first field of an Executive command form, if one existed for this run file.

[Parameter 2] through *[Parameter 16]*

Default: None

Continue to enter each parameter value in order.

Example

If you fill in the **Run** command form as shown below, the Executive executes the command as if you had given the **Files** command. The *[Parameter 1]* field shown below, corresponds to *[File list]*, which is the first parameter field in the **Files** command form; *[Parameter 2]* corresponds to *[Details?]*, which is the second field in the **Files** command form.

```
Run
 Run file          [Sys]<Sys>Files.run
 [Case]
 [Command]
 [Parameter 1]     [d1]<Work>ThisFile
 [Parameter 2]     yes
 [Parameter 3]
    .
    .
    .
 [Parameter 16]
```

If you need to specify wild cards in one of the parameter fields, enclose the file specification in single quotation marks ('), as shown in the following example:

```
Run
 Run file          [Sys]<Sys>Files.run
 [Case]
 [Command]
 [Parameter 1]     '[Sys]<Sys>*'
 [Parameter 2]     yes
 [Parameter 3]
    .
    .
    .
 [Parameter 16]
```

Screen Setup

Use the **Screen Setup** command to change one or more screen attributes of the Executive. Some screen attributes take effect only for the Executive; others remain in effect when you start other applications. See the documentaion for individual applications for more information.

Some screen attributes can be set in *[Sys]<Sys>Config.sys*. See "Configuring Workstation Operating Systems," in the *CTOS System Administration Guide*.

This command is implemented with the Executive internal value !17.

Command Form

```
Screen Setup
  [Reverse video?]                                  _____
  [Large characters?]                               _____
  [Screen length]                                   _____
  [Suppress character attributes?]                  _____
  [Suppress pause between pages?]                   _____
  [Foreground color(s) (current)]                   _____
  [Screen timeout in minutes (current)]             _____
  [Background color (current)]                      _____
  [Suppress blinking cursor? - B27-LCW only]        _____
  [Block for cursor shape? - B27-LCW only]          _____
```

Parameter Fields

[Reverse video?]

Default: No

Enter **Yes** to reverse the normal video characteristics of the monitor. On most monitors, reverse video displays dark characters on a light background. On some high-resolution monitors, however, reverse video displays light characters on a dark background.

If you enter **No** or leave the field blank, normal video characteristics are retained.

[Large characters?]

Default: No

This field applies only to workstations that support both large and narrow character widths. Enter **Yes** to display large characters. If you enter **No** or leave the field blank, narrow characters are displayed.

[Screen length]

Default: No change

This is a dual-purpose field. It controls the screen-length setting on all monitors, as well as the resolution setting on monitors with high-resolution capacity.

To set screen length, enter the number of lines for the display, which includes the status area, the command line, and the area below the command line. The minimum value is 10. The number of lines for a full-screen display varies from monitor to monitor.

Caution

Setting a short screen length can result in Error 2550 (form is too long) when a command form that is longer than the display. Logging off and then signing on again resets the monitor and corrects this condition.

To set high resolution, on monitors that support it, enter **38**. To set low resolution, **29**.

Note: *High resolution cannot be implemented while Window Services is installed.*

[Suppress character attributes?]

Default: No

This field applies only to character-mapped monitors.

Enter **Yes** to suppress command-form highlights when using the Executive. If you enter **No** or leave this field blank, the screen retains the usual character attributes.

[Suppress pause between pages?]

Default: No

Enter **Yes** to suppress the following prompt;

```
Press NEXT PAGE or SCROLL UP to continue
```

This prompt occurs before output is scrolled off the screen. When you do not want to interact with a command, you can suppress the prompt. For example, if you want to leave a tape backup unattended (perhaps overnight), you would suppress the pause between pages.

If you enter **No** or leave this field blank, the Executive prompts you to press **NEXT PAGE** or **SCROLL UP** before information is scrolled off the screen. This is necessary to examine lengthy command output, for example, a long list of files displayed by the **Files** command.

[Foreground color(s) (current)]

Default: No change

This field applies to color monitors only. You can set two foreground colors, one for full intensity and one for half-intensity color, using one of the following methods.

Method 1

For a single foreground color, enter **green**, **white**, **blue**, **yellow**, **amber**, **darkblue**, **purple**, **pink**, **aqua**, **magenta**, **lavender**, **coral**, or **red**.

For two foreground colors, enter the full-intensity color followed by the half-intensity color, separated by a space, for example:

```
[Foreground color(s)]  yellow red
```

Method 2

To create colors, enter **r** for red, **g** for green, and **b** for blue, followed by a number from **0** to **3** to specify the intensity of each color, for example, **rgb013**. The combination of intensities you enter determines the color you see on the screen.

For two foreground colors, enter the full-intensity color followed by the half-intensity color, separated by a space. For example:

```
[Foreground color(s)]  rgb333 rgb013
```

On workstations that support background color, you can enter **rgb000** to produce black characters on the background color you specify (see the *[Background color]* field, below).

See also the *CTOS Editor User's Guide*, for information about setting color with this method, using an interactive display in the Editor application.

[Screen time out in minutes (current)]

Default: No change

This feature automatically turns off the screen when no characters have been typed within a specified number of minutes. Enter the number of minutes to elapse between the time the keyboard is last used and the screen is turned off.

To turn the screen back on, press any key.

To set the screen to always stay on, enter **0**.

This attribute can also be specified in *[Sys]<Sys>Config.sys*, the operating system configuration file, or in the user configuration file. See "Configuring Workstation Operation Systems" and "User File Options for Standard Software," respectively, in the *CTOS System Administration Guide*.

[Background color (current)]

Default: No change

On workstations that support background color, select the background color as described above, for the *[Foreground color(s)]* field. To remove the background color, which changes the background color back to black, enter **rgb000**.

[Suppress blinking cursor? - B27-LCW only]

Default: No

On B27-LCW workstations only, enter **Yes** to prevent the cursor from blinking.

On B28, B38, and B39 character-mapped workstations, this attribute can be set in *[Sys]<Sys>Config.sys*, the operating system configuration file.

[Block for cursor shape? - B27-LCW only]

Default: No

On B27-LCW workstations only, enter **Yes** to change the cursor shape from an underscore character to a box. If you enter **No** or leave this field blank, the underscore cursor is retained.

On B28, B38, and B39 character-mapped workstations, this attribute can be set in *[Sys]<Sys>Config.sys*, the operating system configuration file.

Scrub

Note: *This command is supported on protected mode workstations only.*

Use the **Scrub** command to permanently delete a file or list of files from a volume. The **Delete** command deletes only the pointers to file data; the actual data still remains on the disk and can be recovered in some cases (see the **Undelete** command, later in this section). In contrast, **Scrub** actually destroys data by writing zeros to every byte of the specified file. Scrubbed files cannot be recovered with the **Undelete** command.

This command is implemented with the Executive internal value !24.

Caution

This command permanently destroys data.

Command Form

```
Scrub
  File list
  [Confirm each?]
```

Parameter Fields

File list

Enter the file specification(s) of the file(s) you want to scrub. You can use wild cards or a at-file in this field.

[Confirm each?]

Default: No

Enter **Yes** to be prompted to confirm the scrub operation for each specified file. If you enter **No**, or leave the field blank, you are not prompted for confirmation.

Selective Archive

Use the **Selective Archive** command to back up individual files or directories to tape, floppy diskettes, or another hard disk. **Selective Archive** allows you to back up selected groups of files and requires read access only to the files being archived. **Volume Archive**, in contrast, requires volume-level or device-level access.

You can use **Selective Archive** to perform the following functions:

- Back up individual files or directories
- Back up only files modified on or after a specified date (incremental backup)

Before you can perform tape backups, the Sequential Access Service must be properly installed. See the **Install Sequential Access Service** command, earlier in this section, and "Installing System Services," in the *CTOS System Administration Guide.*

For more information about backing up and restoring data, see also the **Volume Archive** and **Restore Archive** commands, in this section, and "Backing Up and Restoring Data," in the *CTOS System Administration Guide.*

This command is implemented with the run file *SelectiveArchive.run*, command case SA.

Command Form

```
Selective Archive
 File list                                  _____
 [Incremental from]                         _____
 [Confirm each?]                            _____
 [Archive dataset ([QIC])]                  _____
 [Delete existing archive dataset?]         _____
 [Print file]                               _____
 [Verify write?]                            _____
 [Suppress user interaction?]               _____
```

Parameter Fields

File list

Enter a list of the files you want to back up. The list can include single files, directories, sets of files using wild cards, or an at-file.

Note: **Selective Archive** *opens files for read-access only. Therefore, it does not read passwords, nor does it write them to the archive dataset. Use the* **Volume Archive** *command if you want passwords preserved on the archive dataset.*

[Incremental from]

Default: See below

Enter a date. Only eligible files (as specified in the *File list* field) that have been created or modified on or after the specified date are backed up. If no date is specified, all eligible files are backed up.

Optionally, specify a time after the date, as shown in the following example:

May 8, 1991 10:00 AM

If no time is specified, all files from 12:00 midnight on the specified date are backed up.

[Confirm each?]

Default: No

If you enter **Yes**, you are prompted for confirmation before each file is backed up.

[Archive dataset]

Default: *[QIC]*

If you are backing up to floppy diskettes or another hard disk, enter a file specification for the archive file, as shown in the following example:

[Volume]<Dir>d0Archive

A two-digit sequence number, beginning with .01, is automatically appended to the archive file name and is incremented by one for each subsequent diskette.

If you are backing up to tape, enter the tape drive name, for example, *[QIC]*. Tape drive names are defined when you install the Sequential Access Service; see the **Install Sequential Access Service** command, earlier in this section.

Tape mark specifications, which can be included in this field, are described below. If you do not specify a tape mark, the default is 0.

0	Specifies the beginning of the tape.
+	Appends data beyond the last tape mark that already exists on the tape. This is convenient when you do not know exactly how many datasets exist on the tape and you want to avoid overwriting data.

[Delete existing archive dataset?]

Default: No

If you enter **Yes**, an existing archive dataset on the backup medium is automatically overwritten.

If you enter **No** or leave the field blank, you are informed if an archive file already exists, and prompted to press **GO** to overwrite it.

[Print file]

Default: Screen only

Enter a file or printer specification, if you want to write command output to a file or send it to a printer. If the file you specify does not exist, it is created. If it already exists, command output is appended to it.

If you leave this field blank, output is sent to the screen only.

[Verify write?]

Default: No

This field applies only when you archive data to disk backup media, such as floppy diskettes. It does not apply to tape backups. Enter **Yes** if you want to verify that the data written to the archive file matches the data you are backing up. If data does not match, an I/O error (301) is reported. If you enter **No** or leave this field blank, verification is not performed.

[Suppress user interaction?]

Default: No

Enter **Yes** to suppress messages that prompt for user interaction, such as mounting the archive medium. When user interaction is suppressed, **Selective Archive** exits with an appropriate error message, rather than pausing for correction of the condition that is preventing it from continuing.

If you enter **No** or leave this field blank, **Selective Archive** pauses and prompts for user interaction, such as mounting the archive medium or putting the tape drive online.

Server Install

Use the **Server Install** command to start the **Installation Manager** command and to display the list of software applications available for installation from the server.

To be installable from the server, an application must have been publically installed from distribution media (diskette or tape) onto the server. In addition, it must have been installed from a new-style Installation Manager script, which includes it in a database of public software that is available on the server.

The command is implemented with the run file *InstallMgr.run*, command case IS.

Command Form

```
Server Install
  [Non-interactive?]      _____________________________
  [Append to log file?]   _____________________________
```

Parameter Field

[Non-interactive?]

Default: No

Enter **Yes** to perform the installation in noninteractive mode. The installation does not pause for user input, and default parameters from the user configuration file are used.

If you enter **No** or leave this field blank, the command runs in interactive mode. You can then choose to display and change installation defaults, as you would with the Installation Manager command.

[Append to log file?]

Default: No

Enter **Yes** if you want to append to, rather than overwrite, an existing log file. If you enter **No** or leave this field blank, the installation log file is overwritten with each successive installation procedure.

Set Directory Protection

Use the **Set Directory Protection** command to add, change, or remove a directory password, and to change the default file protection level for new files that are created in the directory.

See "File System Security," in Section 2, for information about file protection levels and passwords. See also "Implementing System Security," in the *CTOS System Administration Guide*.

This command is implemented with the Executive internal value !23.

Command Form

```
Set Directory Protection
 Directory name(s)                          _____________
 [Volume or directory password]             _____________
 [New file protection level (current)]      _____________
 [New directory password]                   _____________
```

Parameter Fields

Directory name(s)

Enter the name(s) of the directories for which you want to set the protection level.

[Volume or directory password]

Default: Currently active password

Enter the volume password or current directory password.

[New file protection level]

Default: No change

Enter the default protection level for new files created in the directory.

[New directory password]

Default: No change

If you want to change the directory password, enter the new one here. Passwords can be up to 12 characters long. To remove a directory password, enter a pair of single quotation marks (') in this field.

Set File Prefix

Use the **Set File Prefix** command to add a file prefix to the default path setting. This file prefix is added to the beginning of file names when using the default path. The file prefix is ignored when the path setting is overridden with a volume or directory in the file specification. For example, if you enter *FileName* only in a command form, the default file prefix is placed in front of *FileName*. If, however, you enter *<Dir>FileName* in a command form, the file prefix is ignored.

The Executive status area 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 log out or remove it.

This command is implemented with the Executive internal value !9.

Command Form

```
Set File Prefix
 Default file prefix _____________________________
```

Parameter Field

Default file prefix

Enter the character string you want to use as a file prefix.

To remove a file prefix from the path setting, enter a pair of single quotation marks (') in this field.

Set Mouse Controls

Use the **Set Mouse Controls** command to set the speed with which the cursor tracks the motion of the mouse. On a graphics workstation, this command also allows you to select either a graphics cursor or a character cursor.

Set Mouse Controls has no command form. To execute it, type the command name, then press **GO**.

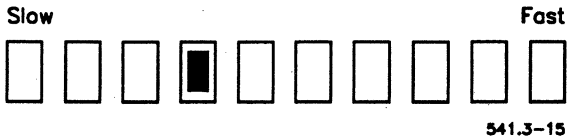
This command is implemented with the run file *SetMouse.run*.

Speed Setting

When you start the **Set Mouse Controls** command, a relative scale, from slow to fast, appears on the screen. The default speed is 4, as shown in Figure 3-15. To change the speed of the mouse cursor, click a different box on the scale.

To exit the display (described below) and save the mouse settings, press **FINISH**.

Figure 3-15. Set Mouse Controls Display



Tracking Method

In addition to a speed setting, you can select either fixed or accelerated cursor tracking, as described below.

F3 (Fixed)	Press this key to select fixed tracking. With this method, the cursor moves only a certain distance in a given length of time.
F4 (Accel)	Press this key to select accelerated tracking. With this method, the cursor moves according to the speed at which you move the mouse.

To demonstrate the difference, select a low speed and fixed tracking. Move the mouse and note cursor movement. Then, switch to accelerated tracking and note cursor movement for the same speed setting.

You can change speed and tracking methods as many times as necessary before exiting the command. To return to the default speed, press **CANCEL**.

Cursor Selection

On graphics workstations only, you can select the shape of the mouse cursor, as described below.

F1 (Graphic)	Press this key to select an arrow-shaped cursor.
F2 (Char)	Press this key to select a box-shaped cursor.

Set Protection

Use the **Set Protection** command to assign a new protection level and, optionally, a file password to the specified list of files.

See "File System Security," in Section 2, for information about passwords and protection levels. See also "Implementing System Security," in the *CTOS System Administration Guide.*

This command is implemented with the Executive internal value !10.

Command Form

```
Set Protection
 File list                    _____________________________
 New file protection level    _____________________________
 [New password]               _____________________________
 [Confirm each?]              _____________________________
```

Parameter Fields

File list

Enter the list of files for which you want to change the protection level.

New file protection level

Enter the protection level you want to assign.

[New password]

Default: No file password

If the specified protection level supports file-level passwords, enter a password for the specified files. This field assigns a file password only. Passwords can have a maximum of 12 characters. To remove a file password, enter a pair of single quotation marks (') in this field.

[Confirm each?]

Default: No

Enter **Yes** to be prompted for confirmation before the protection level is set for each file. If you enter **No** or leave this field blank, you are not prompted for confirmation.

Set Time

Use the **Set Time** command to set the system clock. In a cluster configuration, this command sets the system clock for every workstation connected to the cluster.

This command is implemented with the run file SetTime.run.

Note: *Access to this command can be granted or denied on a per-user basis. See the* **Install Command Access Service** *command, earlier in this section, and "Limiting Access to Certain Commands," in the* CTOS System Administration Guide, *for more information.*

Command Form

```
Set Time
 Date/Time - e.g., Fri May 31, 1991 8:00 am _________
```

Parameter Fields

Date/Time - e.g., Fri May 31, 1991 8:00 am

Enter all or part of the date and time you want to set. For example, to set the time, you only need to enter the time; to change the date, you only need to enter the date. Options for these categories are explained below:

Day of the Week

This entry is optional but appears in the Date/Time status area of the screen. (The correct day of the week is automatically calculated for any specified date.) The day of the week is entered as a three-character abbreviation: Mon, Tue, Wed, Thu, Fri, Sat, Sun.

Date

Enter the date in either of the following formats:

May 8, 1991 or 5/8/91

The month name is entered as a three-letter abbreviation: Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec.

Day of the Month	Enter the day of the month as a number between 1 and 31.
Year	Enter the year with either two or four digits, for example, 1991 or 91.
Time	The time is the current hour and minute and, optionally, 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 change a qualifier by entering the correct qualifier in the command form. You can also use a 24-hour clock, for example, 09:21 is 9:21 AM, and 14:22 is 2:22 PM.

The system has an internal calendar to ensure that entries are valid and consistent. A message informs you if a date/time entry is invalid. For example, since May 31, 1991 is a Sunday, the system does not accept Mon May 31, 1991. It also does not accept Apr 31, Jun 31, Sep 31, or Nov 31. It accepts Feb 29 in leap years only.

Sort

Use the **Sort** command to sort data records according to embedded sort keys. See the *CTOS Sort/Merge Programming Reference Manual* for a detailed description of the command form and parameter fields.

Command Form

```
Sort
 Input file(s)                ____________________________
 Output file                  ____________________________
 Keys                         ____________________________
 [Stable sort?]               ____________________________
 [Work file 1]                ____________________________
 [Work file 2]                ____________________________
 [Print file]                 ____________________________
 [Suppress confirmation?]     ____________________________
```

Sort Public Procedure Names

Use the **Sort Public Procedure Names** command to read an object module and determine if procedure names are listed in the order in which they occur in the module. If procedure names are not so listed, this command sorts them correctly. To run on real mode systems, correct ordering of procedure names is required.

The **Sort Public Procedure Names** command can also be used to sort procedure names in all object modules within a library.

This command is implemented with the run file *PublicSymbolUtility.run*, command case SP.

Command Form

```
Sort Public Procedure Names
 Input object or library      _____________________________
 [Output object or library]   _____________________________
 [List file]                  _____________________________
```

Parameter Fields

Input object or library

Enter the file specification of the object module or library you want to sort.

[Output object or library]

Default: See below

Enter a file specification for the sorted output. If you leave this field blank, output is written to the file specification of the input file, and the original input file is renamed with the suffix *-old*.

[List file]

Default: InputFile.*lst*

Enter a file specification for the object-module list file. The default is the input file specification with the suffix *.lst*.

Spooler Status

Note: *This command is for use with pre-Generic Print System (GPS) printing only. With GPS printing, use the* **Print Manager** *command instead. See the* CTOS Generic Print System Administration Guide *for information about both GPS and pre-GPS printing. See* Using the Print Manager *for information about the* **Print Manager** *command.*

Use the **Spooler Status** command to display information about printers and printer scheduling queues. It also provides the following functions, which are selected from the **Spooler Status** display:

Align Form
Cancel Print
Delete Print Request
Enter Password
Free Printer Channel
Halt Printer
New Printer
Print File
Restart Printer
Select Printer
Select Queue

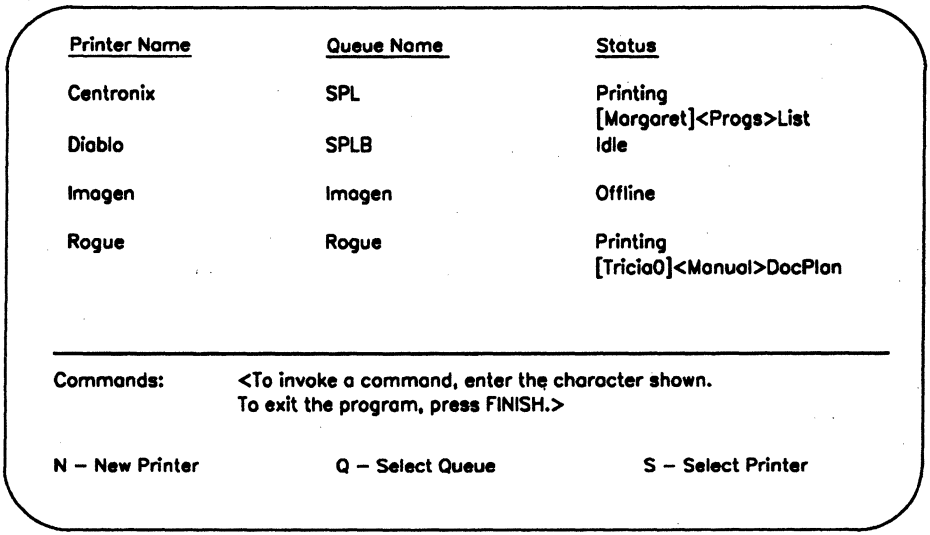
Spooler Status has no command form. To execute it, type the command name, then press **GO**. To exit **Spooler Status**, press **FINISH**.

This command is implemented with the run file *SpoolerStatus.run*.

Main Status Display

After executing the **Spooler Status** command, the main status display appears, as shown in Figure 3-16. The main status display lists information about spooled printers in the cluster.

Figure 3-16. Spooler Status Main Display



541.3-16

From this menu, you can select the following options.

New Printer	Places the specified printer under control of the printing spooler.
Select Printer	Displays a detailed status of the selected printer and provides access to twelve additional options.
Select Queue	Displays the files queued for printing in the selected queue.

New Printer Option

To select *New Printer* from the main status display, type **n**. The following form appears.

```
New Printer
 Printer channel                ____________________
 Printer name                   ____________________
 Queue name                     ____________________
 Printer configuration file     ____________________
 [Priority]                     ____________________
 [Suppress banner?]             ____________________
```

Parameter fields are described below. After filling in the form, press **GO** to execute the command.

Printer channel

Enter the single-character code (shown below) for the printer channel to which the printer is connected:

0	Parallel printer channel
a	Channel A
b	Channel B

Printer name

Enter a name for 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 defines parameters for the printer and is created with 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 to 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 affect the interactive application system.

Note that this priority is not the same as the priority the user selects when issuing a print request. See the **Print** command, earlier in this section.

[Suppress banner?]

Default: No

If you enter **Yes**, the banner page is not printed at the beginning of the file.

If you enter **No** or leave this field blank, the banner page is printed at the beginning of each file.

Select Printer Option

To select *Select Printer* from the main status display, type **s**. The following form is displayed:

```
Select Printer
 Printer name
```

In the *Printer name* field, enter the name of a printer, then press **GO**. The printer display is pictured in Figure 3-17. From this screen, you can select additional options, which are described later. Use **NEXT PAGE** and **PREV PAGE** to scroll through the status display. To return to the main status display, type **m**, then press **GO**.

Figure 3-17 Spooler Status Printer Display

```
Printer:                Centronix
Status:                 Printing
                        [Margaret]<Progs>List
Printer Description:    Parallel
                        Standard forms
Configuration file:     [Sys]<Sys>SplConfig.sys
Location:               Main lab

Queue:                    Spl
Served by:              Centronix

Files Queued

[Master1]<Sys>MonthlyStatus
[TBD]<Schedules>All.list
<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
```

541.3-17

Select Queue Option

To select *Select Queue* from the main status display, type **q**. The following form is displayed:

```
Select Queue
 Queue name
```

In the *Queue name* field, enter the name of a printing queue, then press **GO**.

The queue display is pictured in Figure 3-18. From this screen, you can select additional options, which are described later. Use **NEXT PAGE** and **PREV PAGE** to scroll through the status display. To return to the main status display, type **m**, then press **GO**.

Figure 3-18. Spooler Status Queue Display

```
Queue:                                  SPL
Served by:                              Centronix
Files Queued                            Priority
[Master1]<Sys>MonthlyStatus             0
[Tricia0]<Minutes>11/20.docs            2
[Boss]<Work>Programs.list               5
[TBD]<Schedules>NewProducts             5
[Master1]<Sys>QRS                       5
[June]<Bug>Spray.can                    5
[Alan]<Weasels>Frobozz                  5
[Margaret]<Recipes>ChocChip.cookies     9

<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                              Q – Select queue
M – Main status display                 S – Select printer
P – Print file
```

541.3-18

Printer and Queue Options

You can select additional options from the printer and queue displays. To select an option, type the character shown for it. Then fill in the form (if one is displayed), and press GO.

Options are described in alphabetical order below.

A (Align Form)	This option prints a single page and then pauses for you to align a preprinted form. To print an entire job, after you have aligned the form, use the *Restart* option, described below.
C (Cancel Print)	This option cancels the job that is currently printing.
D (Delete Print Request)	This option deletes a queued print request.
E (Enter Password)	This option allows you to enter a password for the file you want to print. The password must be entered at the workstation to which the printer is attached.
F (Free Printer Channel)	This option frees a printer channel from control of the printing spooler.
H (Halt Printer)	This option pauses a printer. Use the *Restart* option (described below) to restart it.
M (Main Status Display)	This option returns to the main status display (see Figure 3-16).
P (Print File)	This option adds a print request to a scheduling queue. Its fields are identical to those of the *Print* option (described earlier in this section).
R (Restart Printer)	This option restarts a paused printer.

STAT

Use the **STAT** command to access functions provided by the Performance Statistics Service (PSS). **STAT** stands for Statistics and Timing Analysis Tool. It provides a user interface for collecting and viewing statistical performance data. Two key factors of system performance are file system activity and the status of system processes, such as ready, waiting, and so on. Statistics about those factors are helpful to programmers and system administrators in customizing the operating system, distributing disk access activities, and reducing disk errors and fragmentation.

Categories of statistics you can collect and analyze include the following:

- Process and task information
- Disk statistics
- File system statistics

STAT uses the XVT™ window interface, from which options are chosen with the keyboard or a mouse. See Section 2, "Features," for detailed information about using the window interface.

Before you can use a window-interface command, the XVT/CH System Service must be installed. See the **Install XVT/CH Service** command in the Software Release Announcement for that product.

In addition, before you can collect data with the **STAT** command, the Performance Statistics Service must be installed. See the **Install Statistics Service** command, earlier in this section. See also the *CTOS Programming Guide* for detailed information about the Performance Statistics Service.

This command is implemented with the run file *STAT.run*, command case 00.

Command Form

```
STAT
  [Configuration file name(s)]                    _____
  [Maximum processes in ready queue (12)]         _____
  [Maximum partitions (24)]                       _____
```

Parameter Fields

[Configuration file name(s)]

Default: None

Enter the file specification of a configuration file defining a **STAT** session to open. You can specify more than one configuration file and wild-card characters are permitted.

Configuration files are created with the *Session-Definition* option of the **STAT** command. Entering configuration file names in this field opens the defined sessions immediately when **STAT** is invoked.

[Maximum processes in ready queue (12)]

Default: 12

This and the following field determine how much memory is required for storing data returned by the Performance Statistics Service. Enter the maximum number of processes that can be waiting to be processed for use by the **STAT** command.

[Maximum partitions (24)]

Default: 24

This and the preceding field determine how much memory is required for storing the data returned by the Performance Statistics Service. Enter the maximum number of partitions that will be created for use by the **STAT** command.

Main Menu Bar

After you execute the **STAT** command, the window-interface main menu bar appears. It contains the following selectable items:

- File
- Session-Definition
- Operate
- Display
- Help

When you select from the menu bar, a pull-down menu is displayed. Pull-down menus and their subsequent functions are described below.

File Menu

The *File* pull-down menu provides options for setting the path, listing and deleting files generated by **STAT**, and for exiting the application. When you select this pull-down, the following options are displayed:

Set Path

Use this option to set the default path, including node, volume, and directory names, as well as file prefix, and password. All fields are optional. (See the **Path** command, earlier in this section, for more detailed information.)

List/Delete

Use this option to list all files that **STAT** has generated in the specified directory. The specified path and password are shown at the top of the window; you can change them, and then select the **List** pushbutton, to list files in another directory. You can also delete any of the displayed files with the **Delete** pushbutton. See "Entry Fields," below, for a description of the fields displayed in this window.

Exit

Use this option to close all open PSS sessions and to exit the **STAT** command.

Session-Definition Menu

The *Session-Definition* pull-down menu provides options for creating a configuration file that defines a PSS session and for modifying existing session definitions. When you select this pull-down, the following options are displayed:

Create

Use this option to create a configuration file that defines a PSS session. When you select it, additional windows, radio buttons, and entry forms are displayed, as described below.

Statistics

Choose *Statistics* to monitor statistical categories that you select. When you select *Statistics*, another window is displayed, from which you select an operation mode from the following radio buttons:

- *Interactive*. This mode allows you to specify further options, such as gathering data or resetting PSS counters, from the Operate pull-down menu (described later in this section).
- *Noninteractive*. This mode allows you to specify the interval and number of iterations for which to collect PSS data. See "Entry Fields," later in this section.

After you select an operation mode, you select a statistics category from the displayed selection list. Further items of interest are then displayed as check boxes in another window. See "Statistics Categories," later in this section, for a description of each category and its items of interest.

Logging

Choose *Logging* to monitor either active processes in the ready queues or memory usage, which you select with additional radio buttons. When you select *Logging*, an additional entry form is displayed. See "Entry Fields," later in this section, for a description.

Modify

Use this option to modify previously created session definitions. It displays a list of configuration files. When you select one, the window for the *Create* option is displayed, with radio buttons and lists set for the current selection. You can then make changes to the configuration file, as described above.

Operate Menu

The *Operate* pull-down menu provides options for opening and closing PSS sessions, gathering PSS data, and resetting PSS counters. When you select this pull-down, the following options are displayed:

Open Session

Use this option to open a session based on the definition contained in a configuration file. You may choose a configuration file from the currently displayed list, or you may change the path and select the **List** pushbutton to display a new list files.

Gather Data

Use this option to gather data from an open interactive statistics session. When you choose a session from the displayed list, the following radio buttons are displayed:

All Categories	Choose this button to gather data for all displayed categories (which you previously selected with the *Session-Definition* option).
Individual Categories	Choose this button to gather data for categories you select from the displayed list.

When you gather data, a heading lists the session name, starting time, gathering time, and other relevant information about the session. Statistics that are gathered are displayed in a scrollable window following the heading. You can save gathered statistics in a file with the **Save** pushbutton (see also its description under "Pushbuttons," below).

Reset Counters

Use this option to reset all counters in an interactive statistics session or the counters in a specified subset of counters. Choose the session for which to reset counters from a list of open sessions.

Close Session

Use this option to select a session to close from a list of open sessions.

Display Menu

The *Display* pull-down menu provides options for monitoring the status of open sessions and displaying the data returned by PSS. When you select this pull-down, the following options are displayed:

Opened Sessions

Choose this option to display a list of open sessions.

Data File

Use this option to display a list of files containing collected data to view that data. When you open a data file, a heading describes the type of session and other relevant information. Statistics contained in the file are displayed in a scrollable window following the heading.

Help Menu

The *Help* pull-down menu is provided by the XVT/CH System Service. It provides information about the window interface, **STAT** functions, and PSS error codes. See your XVT documentation for information about using this help facility.

Entry Fields

The following entry fields are present throughout the **STAT** command and are described in alphabetical order below:

[Device Spec.]

To collect statistics from another node where the Performance Statistics Sservice is installed, specify the node name in the form of *{Node}*. To collect statistics from the server, specify *{Server}*.

Interval in Sec.

Specify the number of seconds between each iteration.

Log Heap Size in 512 Bytes

Specify a size for the log heap, in 512 byte increments. The log heap is an area of memory required by the Performance Statistics Service.

Number of Iterations

Specify the number of iterations you want to monitor.

[Password]

Specify the password for the path you enter.

[Path]

Specify a path, which can include node, volume, directory, and file prefix.

Session Name (Max. 12)

Specify a name containing 12 characters or less, for the session-name configuration file. The suffix *.sta* will be appended for a statistics session, or *.log* for a logging session. Additionally, to form the name of a data-collection file for the session, the date, time, and *.data* are appended to the name of configuration file (see also the **Save** pushbutton below).

Pushbuttons

The following pushbuttons are available when applicable, throughout the **STAT** command:

Cancel	Exits the currently displayed window.
Delete	Deletes **STAT**-related files from a list of displayed file names.
Help	Displays help for the current window.
List	Displays a list of items. It is frequently activated after changing the path setting to display a list of files.
Next	When creating a configuration file, displays the next device name in sequential order from *f0* to *f3*, and then *d0* to *d17*. When viewing noninteractive statistics or logging data, displays the previous iteration.
OK	Confirms a choice.

Previous	When creating a configuration file, displays the previous device name in reverse order from *d17* to *d0*, and then *f3* to *f0*. When viewing noninteractive statistics or logging data, displays the next iteration.
Remove	Removes an item from the list of those selected.
Save	Saves gathered data in a file. See the *Session Name* entry field for a description of file names assigned by **STAT**.
Select	Selects a highlighted item, usually from a list.
Show	Displays a list of items currently selected.

Statistics Categories

The following categories are displayed in a selection list when you define a statistics session. After choosing one, a set of check boxes is displayed, from which you select items of interest for the category.

Process/Task Info

Choose this category to collect information about tasks and processes managed by the operating system. This category is further broken down in the following items of interest:

Idle Process Cycles	Accumulated number of invocations of the operating system null process.
Normal Task Switches	Accumulated number of normal task switches, including interrupts.
Processes Terminated Normally	Accumulated number of processes terminated normally.
Processes Terminated Abnormally	Accumulated number of processes terminated abnormally, such as the termination caused by **ACTION-FINISH** or by ErrorExit with error.
Partitions Swapped In	Accumulated number of partitions swapped into memory.

Partitions Swapped Out	Accumulated number of partitions swapped out of memory.

Disk Statistics

Choose this category to collect information about mass input/output (I/O) activities. This category is further broken down in the following items of interest:

Hard Disk Errors	Accumulated number of unrecoverable errors on the specified devices.
Soft Disk Errors	Accumulated number of recoverable errors on the specified devices.
IOB's in Process	Number of active I/O blocks at the moment of inquiry.

File System Statistics

Choose this category to collect information about file system activity. This category is further broken down in the following items of interest:

Files Opened	Accumulated number of files opened for the specified devices.
Files Closed	Accumulated number of files closed for the specified devices.
Files Created	Accumulated number of files created for the specified devices.
File Deleted	Accumulated number of files deleted for the specified devices.
File Renamed	Accumulated number of files renamed for the specified devices.
File Length Changes	Accumulated number of files with changed file length for the specified devices.
Remake File Handles	Accumulated number of files handles remade for the specified devices.

Disk Seeks (READ and WRITE)

Choose these categories to collect the accumulated number of disk seeks required to read or write blocks of data ranging from one sector to 128 sectors for the specified devices. You are prompted to select a device and to enter a sector number (or a range of sector numbers).

Virtual Disk Access (READ and WRITE)

Choose these categories to collect the accumulated number of virtual disk extents required for the read or write requests issued for the specified devices. You are prompted to select a device and to enter a sector number (or a range of sector numbers).

Logical Reads and Logical Writes

Choose these categories to collect the accumulated number of logical read or write requests issued for the specified devices. You are prompted to select a device and to enter a sector number (or a range of sector numbers).

Status Code Reporter

Use the **Status Code Reporter** command to obtain a description of specified error codes. You can also search for character strings in the status code descriptions.

This command is implemented with the run file *Erc.run*.

Command Form

```
Status Code Reporter
  [Error code(s)]                 _____
  [Range of error codes]          _____
  [Print file]                    _____
  [String(s) to search]           _____
  [Error message file name]       _____
```

Parameter Fields

[Error code(s)]

Default: None

Enter the number of the error code(s) for which you want a description. Values may be either decimal or hexadecimal numbers; end hexadecimal values with the letter "h". Decimal error codes are between 0 and 65535, and hexadecimal error codes between 0h and FFFFh.

[Range of error codes]

Default: None

Enter the range of error codes for which you want descriptions. The range is entered as a starting and ending value, separated by a space. If you enter a single number, descriptions are returned for error codes starting with that number through the last error code. If you enter an asterisk (*) wild card, descriptions for all error codes are returned.

[Print file]

Default: Screen only

If you want a record of command output, enter a file or printer specification. If the file you specify does not exist, it is created. If it already exists, new output is appended to the file.

If you leave this field blank, command output is sent to the screen only. If you want output sent to a file only, specify **NoVid** after the file specification. Separate it from the file specification with a space, as shown in the following example:

```
[Print file]  [Sys]<EList>ErrorFile NoVid
```

[String(s) to search]

Default: No string

To search for character strings within status code descriptions, enter the string(s) for which to search. You can enter a maximum of five strings in this field. Enclose each string in single quotation marks (for example, 'bad heap'). Leave a space between strings.

[Error message file name]

Default: *[Sys]<Sys>StatusCodesMsg.bin*

Enter the file specification of the file containing the status code messages.

Stop Record

Use the **Stop Record** command to stop recording keystrokes in a submit file. See the **Record** command, earlier in this section, for information about recording submit files.

The **Stop Record** command can be executed in either of the following ways while the Executive is recording:

- On the command line, type **Stop Record**, then press GO (there is no command form).
- Press CODE-BOUND.

If you execute **Stop Record** when the Executive is not recording, the following message is displayed:

```
The Executive is not recording.
```

This command is implemented with the Executive internal value !20.

Submit

Use the **Submit** command to execute submit files. It reads command input from a file, rather than from the keyboard.

Submit files are created with the **Command File Editor** and can be edited to contain escape sequences for including variables, pauses for user input, and screen messages. For information about creating submit files, see "Submit File Techniques," later in this section. See also the *CTOS Programming Guide* and the sections on keyboard management in the *CTOS Operating System Concepts Manual.*

This command is implemented with the run file *Submit.run.*

Command Form

```
Submit
 File list                 _____________________________
 [Parameters]              _____________________________
 [Force expansion?]        _____________________________
 [Show expansion?]         _____________________________
 [Continue on error?]      _____________________________
 [Start after]             _____________________________
```

Parameter Fields

File list

Enter the name of the submit file(s) from which characters are read. Files are read sequentially.

[Parameters]

Default: None

Enter the parameter values that will replace variables in the submit file. The first parameter typed in this field replaces all instances of *%0*, the second, *%1*, and so on, through *%9*.

To replace a variable with a null string (that is, to leave it blank), enter a pair of single quotation marks (') as a place holder. To enter a string of characters containing spaces, enclose the string in single quotation marks.

[Force expansion?]

Default: No

Specify **Yes** to force replacement of system-defined and recursive variables. If you specify **No** or leave this field blank, system-defined and recursive variables are not expanded, unless one of the following conditions is met:

- A parameter value for a user-defined variable is specified in the *[Parameters]* field of the command form, or
- The submit file begins with the characters *%'*

For more information about variables and recursion, see "Using Variables," and "Using Recursive Submit Files," later in this section.

[Show expansion?]

Default: No

If you enter **Yes**, expanded variables are shown on the screen before the submit file begins executing. This is particularly useful for debugging complicated conditional expansions. If you enter **No**, the submit file begins executing without displaying expanded variables.

[Continue on error?]

Default: No

If you enter **Yes**, the submit operation continues when an error occurs. If you enter **No** or leave this field blank, the submit operation terminates when an error is encountered.

Note that this parameter applies only to errors that result from commands that are being executed by the submit facility. It does not apply to syntax errors in the submit file.

[Start after]

Default: Current date and time

Enter the date and/or time at which you want to submit file to begin executing, using any valid date/time format (see the **Set Time** command, earlier in this section). If you leave this field blank, the submit operation begins as soon as you press **GO**.

Submit File Techniques

You create executable submit files using Executive commands. The submit facility reads keystrokes from a file, just as if they were typed on the keyboard.

A submit file is an ASCII text file written in a script language that can be interpreted by the Executive. This script language consists of the command names and parameter values you use with the Executive. In addition, it includes escape codes and keyboard literal characters, which perform special functions.

Complex submit files can include the following programming features:

- Pauses for user input
- Specification of system-defined variables
- Specification of user-defined variables
- Execution from Executive command forms
- Discrimination of conditional variables
- Nesting, where one submit file calls another and then returns control to the calling file
- Recursion, where a submit file resubmits itself over and over again

Note: *The* **Record** *command inserts keyboard escape sequences that begin with ¢3 and end with ¢4 into the submit files it creates. Do not change or remove such escape sequences unless you fully understand 12-byte keyboard records, as described in the* CTOS Operating System Concepts Manual.

Creating Submit Files

You can create simple submit files with function keys and the **Record** command; see "Macros," in the *CTOS Executive User's Guide.*

You can create complex submit files with the **Command File Editor** and then edit them with a text editor to add escape sequences, messages, and special characters. Unlike the **Record** command, the commands you output with the **Command File Editor** are not executed while you create the submit file.

To create a submit file with the **Command File Editor**, follow these steps:

1. On the command line, type **Command File Editor**, then press GO.
2. Position the highlight on the first command you want to include in the submit file.
3. Press F9 (Output).
4. Press RETURN to move the highlight to the *Output file* field.
5. Type a file specification for the submit file.
6. Press GO.
7. If a command form is displayed, fill in parameter values or variable escape codes (described later in this section); then, press GO.
8. Repeat steps 2, 6, and 7 for each command you want to include in the submit file.
9. Press FINISH, then GO to save the submit file and exit the **Command File Editor**.

See also the **Command File Editor** command, earlier in this section. Figure 3-19 shows a submit file created with the **Command File Editor**.

Editing the Submit File

After you have created a submit file, you can insert escape codes, messages, and special characters with a text editor. The Editor application works well, because it is designed for editing ASCII files. If you use OFIS Document Designer or some other word processing program, be sure to remove formatting characteristics from the file (see the manual for your word processing program).

Figure 3-19. Submit File Created With the Command File Editor

```
%'
Create Directory
%'New directory name(s)' NewDir
%'[Default protection level (15)]'
%'[Maximum number of files (75)]' 100
%'[Password for new directory]'
%'[Volume password]^'
•%'
Path
%'[Volume]*' d1
%'[Directory]*' NewDir
%'[Default file prefix]*'
%'[Password]^'
%'[Node]*'
•%'
LCopy
%'File list' @Files.fls
%'[File prefix(s) from]' [Sys]<Sys>
%'[File prefix(s) to]'
%'[File suffix(s)]'
%'[Overwrite ok?]'
%'[Confirm each?]'
%'[Continue on error?]'
%'[Verify copy?]'
•%'
```

Inserting Special Characters

Certain keystrokes, such as **GO** and **FINISH**, are represented by special keyboard literal characters. When placed in a submit file, literal characters execute the function for the keystroke, as if it were issued from the keyboard.

Two commonly used keyboard literals are the filled box for **FINISH** and the filled circle for **GO**. See the *CTOS Procedural Interface Reference Manual* for a detailed list of keyboard literals.

Note: *When the keyboard literal for* **F1** *(Redo) is used in a submit file, it reexecutes the last command actually typed on the keyboard, rather than the last command executed from the submit file. The keyboard literal for* **F1** *is the division symbol.*

You can insert keyboard literal characters while editing a submit file with a text editor. To insert them with the Editor, follow these steps:

1. Position the cursor where you want to insert a character.
2. Press **CODE-'** (single quotation mark).
3. Press the key that corresponds to the literal character, for example, **GO** to insert •.

Keyboard literals can also be inserted by pressing **CODE-i**, and then typing the two-digit hexadecimal equivalent for the character. See the *CTOS Procedural Interface Reference Manual* for a table of the standard character set.

Using Escape Codes

An escape code instructs the Executive to temporarily abandon execution of keystrokes and to perform a special function. The following escape codes are recognized by the Executive; each is described in detail, later in this section.

%' or *%"*	Allows you to insert comments. Any text preceded by a percent sign (%) and then enclosed in single (') or double (") quotation marks is a comment and is ignored by the Executive. In addition, beginning a submit file with *%'* ensures proper expansion of system-defined and recursive variables.
¢1char	Pauses execution of the submit file until a designated keystroke is issued by the user. For example, *¢1•* instructs the Executive to stop execution until the user presses **GO**.
¢2	Instructs the Executive to immediately terminate execution of the submit file.
%variable	Instructs the Executive to expand a variable.

Adding Comments

To add comments in a submit file, type the percent sign (%) and then surround commented text with either single (') or double (") quotation marks. If a commented line ends with **RETURN**, the closing quotation mark is not required. The example below shows various styles of commented text.

Note: *Characters shown in boldface type in the submit file examples are for emphasis only. Actual submit files do not contain such formatting characteristics.*

```
%' The following comment lines all are valid:'

%'          These two lines, starting and ending'
%'          with a single quote,'

%'          These four lines, starting with a
%'          single quote and ending with a
%'          RETURN. (No quotation mark is required
%'          when the line ends with a RETURN).

%"          These two lines starting and ending"
%"          with a double quote."

%"          These three lines starting with a
%"          double quote and ending
%"          with a RETURN.

%' These lines, which extend over more than one line on the screen, are also a
valid comment, because there is no RETURN at the end of the first line.

%' Field names are surrounded by quotes, therefore becoming comments. Notice
how the command name itself and parameter values are not preceded by an escape
code. These are not comments and will be interpreted by the Executive when the
submit file is executed.
```

Pausing for User Input

You can cause a submit file to stop in a predetermined place to await input from the user. To do this, you use the *¢1* escape sequence. With this escape sequence, you specify a key for the user to press to reactivate the submit file.

In the following example, the escape sequence shown in bold pauses for the user to enter a password, then restarts when the user presses **GO**. You could, however, specify any other key, such as **RETURN**, **NEXT PAGE**, or a function key. To specify a particular keystroke, you insert a keyboard literal into the submit file, as described earlier in this section.

```
%'
Path
%'[Volume]*' [!Sys]
%'[Directory]*' <Sys>
%'[Default file prefix]*'
%'[Password]^' ¢1•
%'[Node]*'
•
Create Directory
%'New directory name(s)' TestDir
%'[Default protection level (15)]'
%'[Maximum number of files (75)]' 100
%'[Password for new directory]'
%'[Volume password]^'
•
```

Displaying Messages to the User

Once you know how to pause a submit file, you can display messages for users, so that they will know what to do when the submit file stops. Although messages are not escape sequences, they are described here because of their association with the *¢1* escape sequence. To display messages, you use either the **Type** or **Copy** command.

See "Video Techniques," later in this section, for information about using screen and character attributes in submit file messages.

With the Type Command

The following example shows how to add a message with the **Type** command. Notice that you are typing from the device specification *[Kbd]*. The **FINISH** symbol (a filled box) follows output from the keyboard, to return control of the keyboard to the submit facility.

```
%'
Type
%'File list' [Kbd]
%'[Confirm each?]'
●
Enter the password for the server in the Path command
form below. Then, press Go to continue.
■
Path
%'[Volume]' [!Sys]
%'[Directory]' <Sys>
%'[Default file prefix]'
%'[Password]' ¢1●
%'[Node]'
●
```

With the Copy Command

With the **Copy** command, you direct output to the screen *([Vid])*. You can display keystrokes from the submit file using *[Kbd]* (as shown above for the **Type** command), or you can display keystrokes from a file.

The advantage of **Copy** over **Type** is that you can store standard messages in files. For example, the following text could be stored in a file named *[d0]<Messages>Password.msg*:

```
Enter the password for the server in the Path command
form below. Then, press GO to continue.
```

To display the text shown above, you copy *[d0]<Messages>Password.msg* to *[Vid]*, as shown in the following example. Notice that you do not need to type any text for the message or include a **FINISH** symbol in the submit file.

```
%'
Copy
%'File from' [d0]<Messages>PressGO.msg
%"File to' [Vid]
%'[Overwrite ok?]'
%'Confirm each?]'
•

Path
%'[Volume]*' [!Sys]
%'[Directory]*' <Sys>
%'[Default file prefix]*'
%'[Password]^' ¢1•
%'[Node]*'
•
```

Terminating a Submit File

The *¢2* escape code immediately terminates a submit file. It is used with conditional variables, for example, to end a submit sequence if a required parameter value is missing. See "Using Conditional Variables" and "Using Recursive Submit Files," later in this section.

Using Variables

The Executive allows you to include escape codes for variable parameter values in submit files. This allows a submit file to use different parameter values each time it is executed.

For example, suppose a submit file creates a directory in which to copy files. The name of the directory, however, must be different for each user who executes the submit file. Rather than pausing for the user to fill in a directory name, you can insert a variable within the submit file; this variable will then be expanded with an actual parameter value, in this case, a directory name.

The Executive recognizes and expands the following types of variables:

System-defined	The variable is expanded with a parameter value defined by the system.
User-defined	The variable is expanded with a parameter value defined by the user who is executing the submit file.

The following sections describe how to use both system-defined and user-defined variables.

System-Defined Variables

The following variables are expanded with values provided by the system.

%[	Expands to the volume or device name that appears in the path setting.
%<	Expands to the directory name that appears in the path setting.
*%^*	Expands to the password that is in effect when the submit file is executed.
%U	Expands to the user name that appears in the status area of the screen.
%D	Expands to the date and time that appears in the status area of the screen. This variable must be followed by a date/time template, which is described later in this section.

Path Variable

You can specify the default path setting and password by using the path variables. The following example shows how to specify the default volume and directory in a submit file. You can also use the path variables in file specifications.

```
%'
Path
%'[Volume]*' %[
%'[Directory]*' %<
%'[Default file prefix]*'
%'[Password]^'
%'[Node]*'
.
```

Path variables are expanded before the submit file begins executing. Therefore, all path variables (that is, volume, directory, and password) are expanded to the path setting that appears in the status area when the submit command is initiated.

Expanded volume and directory names do not include brackets; therefore, when using volume and directory variables in a file specification, you must include brackets in the submit file, for example:

```
[%[]<%<>FileName
```

The password variable is useful when the user needs to change passwords during a submit sequence, for example, to use a disk on the server. To reenter the password that was in effect when the user started, enter %^ in the *[Password]* field of the **Path** command form.

User Name Variable

You can specify the SignOn user name with the user name variable. This is particularly useful if several users are sharing a disk on the server. The following example shows how to create a directory with the SignOn user name and then path to it.

```
%'
Create Directory
%'New directory name(s)' %U.NewDir
%'[Default protection level (15)]'
%'[Maximum number of files (75)]' 100
%'[Password for new directory]'
%'[Volume password]^'
•
Path
%'[Volume]*'
%'[Directory]*' %U.NewDir
%'[Default file prefix]*'
%'[Password]^'
%'[Node]*'
•
```

The SignOn user name can also be used in file specifications, written as keyboard output to a log file, or in any other instance where it is valid as all or part of a parameter value.

Date/Time Variable

You can specify the system date and time with the *%D* variable. This is a useful feature for writing messages to a log file. For example, the following excerpt from the submit routine for the **Daily Archive** command, shows a message written to the log file named *[Sys]<Sys>Archive.log*. The characters following the date/time variable are the date/time template, which is described below.

```
%'
Append
%'File list from' [Kbd]
%'File to' [Sys]<Sys>Backup.log
%'[Confirm each?]'
●
Begin incremental backup on %D|!Www! !Nnn! !dd!, !yyyy! !tt!:!0m!|■
```

When you use the *%D* variable, you include a date/time template to specify how you want the date to be expanded. A date/time template is enclosed in vertical line (|) delimiters; each entry within the template, that is, day of the week, month, and so on, is then enclosed in exclamation point (!) delimiters.

For example, the following template (as shown in the example above) expands the date and time Monday, April 2, 1988 1:58 PM, as follows:

Template:	*|!Www! !Nnn! !dd!, !yyyy! !tt!:!0m!|*
Expands to:	Mon Apr 2, 1989 13:58

Table 3-5 lists valid entries for date/time templates and explains how they are expanded. The following examples show you how to create date/time templates in a variety of formats. Punctuation, such as commas and slashes, as well as spaces, are placed outside the exclamation point delimiters (!) that define each entry.

Sample Date Format	Date/Time Template to Use
Friday June 5, 1992	*/!W*! !N*! !*d!, !y*!/*
FRIDAY JUNE 5, 1992	*/!*W! !*N! !*d!, !y*!/*
Fri Jun 5, 1992 6:30 pm *or* Fri Jun 26, 1992, 11:55 am	*/!Www! !Nnn! !*d!, !y*! !*h!:!0m! !aa!/*
27 July 1992, 0400 *or* 3 April 1992, 2315	*/!*d! !N*! !y*!, !0t!!0m!/*
5/8/92	*/!*o!!!*d!!!yy!/*
05/08/92	*/!0o!!!0d!!!yy!/*
7/27/92, 6:30 PM	*/!*o!!!*d!!!yy!, !*h!:!0m! !AA!/*

Table 3-5. System Date/Time Template Entries

Date and Time Entries	How They Are Expanded
n	One lower case character of the month.
N	One upper case character of the month. For example, *Nnn* specifies a three-character month abbreviation, such as **Apr**, with the first character capitalized.
w	One lower case character of the day of the week.
W	One upper case character of the day of the week. For example, *Www* specifies a three-character abbreviation, such as **Mon**, with the first letter capitalized.
a	One lower case character of the am or pm designation. For example, *aa* specifies **am** or **pm**, depending on the time of day.
A	One upper case character of the AM or PM designation. For example, *AA* specifies **AM** or **PM**, depending on the time of day.
o	One digit of the month for use in all-numeral dates, such as 10/25/89. For example, *oo* specifies two digits for the month, such as **04** for April. See the asterisk (*) below for information about suppressing the leading zero for one-digit months.
d	One digit of the date of the month. For example, *dd* specifies a date, such as **24**, in Apr 24, 1989.
y	One digit of the year. For example, yyyy specifies a four-digit year such as **1989**; *yy* specifies a two-digit year such as **89**.
t	One digit of the hour of the time of day, using a 24-hour clock.
h	One digit of the hour of the time of day, using a 12-hour clock.
m	One digit of the minute of the time of day.
s	One digit of the second of the time of day.

(continued)

Table 3-5. System Date/Time Template Entries (cont.)

Date and Time Entries	How They Are Expanded
0	Inserts a zero (0) as a place holder if the value for a two-digit entry is a one-digit number. Should always be used to specify minutes. For example, *hh:0m* expands to **12:09**; in contrast, *hh:sm* expands to **12:9**, which is not a valid time setting. Can also be used in all-numeral dates, for example, *0o/0d/yy* specifies **01/04/89**, whereas *oo/dd/yy* specifies **1/4/89**.
*	Expands the entry to contain all possible characters for the field. For example: *W** specifies the full name of a day of the week, in upper and lower case, such as **Wednesday**. **N* specifies the full name of a month, all capitalized, such as **AUGUST.** *y** specifies four digits of the year, such as **1989**. **o* specifies a numeric, such as **4** in 4/11/89, without a place holding zero for one-digit numbers. **d* specifies a numeric, such as **2** in 4/2/89, without a place holding zero for one-digit numbers.

User-Defined Variables

User-defined variables allow the user to enter parameter values that may vary with each execution of the submit file. To do so, you use the % escape character, followed by a number from 0 to 9, for example:

%0

Variables can be used, for example, to allow the user to define a directory name and password, as shown below:

```
%'
Create Directory
%'New directory name(s)' %0
%'[Default protection level (15)]'
%'[Maximum number of files (75)]'
%'[Password for new directory]' %1
%'[Volume password]'
.
```

The same variable can be used more than once within a submit file. The following example shows how the variables %0 and %1 are used to create a directory and then path to it:

```
%'
Create Directory
%'New directory name(s)' %0
%'[Default protection level (15)]'
%'[Maximum number of files (75)]'
%'[Password for new directory]' %1
%'[Volume password]'
.
Path
%'[Volume]*'
%'[Directory]*' %0
%'[Default file prefix]*'
%'[Password]^' %1
%'[Node]*'
.
```

User Defined Parameter Values

User-defined parameter values are entered into the **Submit** command form when the user executes the submit file, as shown in the following example. Notice that parameter values are separated with a space, and yes is entered in the *[Force expansion?]* field.

```
Submit
 File list              [Sys]<Sys>Sample.sub
 [Parameters]           NewDir NewPswd
 [Force expansion?]     yes
 [Show expansion?]
 [Continue on error?]
 [Start after]
```

The Executive handles the user-defined parameter values as follows:

- Each instance of *%0* is expanded with the first parameter value, in this case, **NewDir**.
- Each instance of *%1*, is replaced with the second parameter value, in this case **NewPswd**, and so on.

Note: *To ensure expansion of variables in a submit file, begin the submit file with the %' escape code. When this escape code appears as the first entry in a submit file, all variables are expanded, whether or not the user enters* **Yes** *in the* [Force expansion?] *field of the* **Submit** *command form. When using %' to force expansion, however, no characters, including* **RETURN**, *can precede it.*

Omitting a Variable

At times, the user may not require a parameter value for all variables in the submit file. For example, a submit file could contain variables *%0* through *%2*, as shown in the following example:

```
%'
LCopy
%'File list' %0
%'[File prefix(s) from]' %1
%'[File prefix(s) to]' %2
%'[File suffix(s)]'
%'[Overwrite ok?]'
%'[Confirm each?]'
%'[Continue on error?]'
%'[Verify copy?]'
.
```

To omit the parameter value for a variable, a pair of single quotation marks (') is entered in the command form as a place holder. The following example shows how to omit the parameter value for *%1* in the submit file shown above:

```
Submit
 File list              [Sys]<Sys>Sample.sub
 [Parameters]           File1 '' [d0]<ThisDir>
 [Force expansion?]
 [Show expansion?]
 [Continue on error?]
 [Start after]
```

Place holders are not required if the submit file is being executed with an Executive command. See "Creating Commands for Submit Files," later in this section.

Replacing a Variable With Multiple Values

In a submit file, it is possible to specify an undetermined number of variables for a command form field. To do so, you use the *%>n* escape code (where *n* is the number of user-defined variable). With the greater than sign (>), the variable is expanded to include all parameter values greater than its value. For example, *%>0* is expanded to all parameters entered in the **Submit** command form, except the first.

By using this escape code, it is possible for more than ten variables to be expanded in a submit file. In addition, you do not need to know the exact number of parameters the user will enter. The following example shows how to specify an indeterminate number of entries in the *File list* field of the **Type** command form.

```
%'
Type
%'File list' %0 %>0
%'[Confirm each?]'
.
```

The submit file shown above could be executed with parameter values as shown below. Each parameter is a file name to be expanded in the **Type** command form in the submit file:

```
Submit
 File list              TypeMany.sub
 [Parameters]           File1 File2 File3 File4 File5
 [Force expansion?]
 [Show expansion?]
 [Continue on error?]
 [Start after]
```

Then, when the submit file is executed, the variables (*%0* and *%>0*) are expanded as follows:

```
Type
 File list        File1 File2 File3 File4 File5
 [Confirm each?]
```

Creating Commands for Submit Files

You can create Executive commands and command forms with which to execute submit files. This is particularly useful for submit files containing variables, which require the user to enter parameter values.

In an Executive command form, each field corresponds to a variable in the submit file. The entry in the first field replaces *%0*, the second replaces *%1*, and so on. For example, you could create a command named **Directory Backup**, which deletes *-old* files and performs a selective backup on a single directory; an example is shown below.

```
%'
Delete
%'File list' <%0>*-old
%'[Confirm each?]'
•
Selective Archive
%'File list' <%0>*
%'[Incremental from]'
%'[Confirm each?]'
%'[Archive dataset]' [%1]<Sys>Arc
%'[Delete existing archive dataset?]' yes
%'[Print file]'
%'[Verify write?]'
%'[Suppress user interaction?]'
•
```

The command form for the above submit file would contain two parameter fields, as shown in the following example:

```
Directory Backup
 Directory to back up  _____________________________
 Floppy drive to use   _____________________________
```

When a user executes the command, *%0* in the submit file is expanded with the entry in the *Directory to back up* field; *%1* is expanded with the entry in the *Floppy drive to use* field.

Rules and Restrictions

The following rules apply when you create a command for a submit file:

- The submit file name must end with the suffix *.sub*.
- The command name must match the name of the submit file, excluding the suffix. For example, for the submit file for the command named **Directory Backup** is *DirectoryBackup.sub*.
- The run file for submit file commands is *SubCmd.run*. For the command to be executable from any directory, both the submit file and *SubCmd.run* must be stored in the same directory, in most cases *[Sys]<Sys>*.

Note: Beginning submit files with the characters %' ensures that system-defined and recursive variables will be properly expanded.

How to Create a Command

The following procedure describes how to create a command for a submit file using the Command File Editor.

1. Copy the submit file into *[Sys]<Sys>*.
2. On the command line, type **Command File Editor**, then press **GO**.
3. Press **F2** (Add). The following form is displayed at the top of the screen:

```
Command name      _____________________________________
Copy from command _____________________________________
```

4. In the *Command name* field, type the name of the command you want to create. Remember, it must match the name of the submit file (see "Rules and Restrictions," above).
5. Press **GO**. The command name appears at the top of the screen.
6. Press **RETURN**. A blank highlighted field appears.
7. Enter the parameter field description for the variable *%0* in the submit file; then, press **RETURN**.
8. Repeat Step 7 for each variable in the submit file.

9. When you have finished entering parameter field descriptions, press **CODE-RETURN**. The highlight moves to the *Run file* field at the bottom of the screen.
10. Type **[Sys]<Sys>SubCmd.run**, then press **RETURN** twice.
11. Optionally, type a brief description for the command in the *Help message* field.
12. Press **GO** to create the command.
13. Press **FINISH**, then **GO**, to exit.

Using Conditional Variables

Conditional variables are used to test whether or not a parameter value is present before the submit file executes certain commands. For example, they can be used to display error messages when required parameters are omitted. The Executive recognizes the following conditional variables:

%# "Do if not present"	Instructs the Executive to perform a certain function if a parameter value for the variable is not present.
%? "Do if present"	Instructs the executive to perform a certain function if a parameter value for the variable is present.

The commands or keystrokes to execute if the conditional variable is true are encased in delimiting characters of your choice, after the conditional variable. These are called *escape sequences.*

"Do If Not Present"

An excellent example of "do if not present" (%#) is found in the **Weekly Archive** submit routine, which is supplied with Standard Software. The following excerpt from that submit file shows an escape sequence that is executed if no parameter value is entered for the variable *%0*; this variable corresponds to the mandatory parameter field, *Volume to archive*. The escape sequence displays an error message and terminates the submit file if a parameter value is not entered for the *%0* variable.

```
%'
%#0:
Type
%'File list' [Kbd]
%'[Confirm each?]'
.
Operator error.  No device name was supplied in the command form.
¢2:
```

In the above example, note the following:

1. The %#0 escape code specifies that the command and keystrokes enclosed within delimiters are to be carried out if no user-defined parameter is entered for the variable %0.
2. Colons (:) are the delimiters for the escape sequence.
3. The **¢2** escape code terminates the submit file.

The escape sequence following *%#0* is carried out only if a parameter value is not present; therefore, it is ignored if the user enters a parameter value for the variable.

"Do If Present"

Another example from the **Weekly Archive** submit routine shows the "do if present" (*%?*) conditional variable. The following excerpt shows two escape sequences that are executed if a parameter value is entered for the *%2* variable; this variable corresponds to the optional parameter field, *[Disable cluster?]*. The escape sequences display a message and then disable the cluster.

```
%'
Type
%'File list' [Kbd]
%'[Confirm each?]'
●
                Insert the first archive diskette.
%?2~
                  The cluster will be disabled.
~
                    Press GO when ready.
■
¢1●●
%?2~
Disable Cluster
%'[Time interval in seconds]'
%'[XE line to leave up]'
●~
```

In the above example, note the following:

1. The **%?2** escape codes specify that the commands and keystrokes enclosed within delimiters are to be carried out if a parameter value is entered for the *%2* variable.
2. Tildes (~) are the delimiters for the escape sequences.
3. Portions of the message and the **¢1** escape code are not included in the %?2 conditional escape sequences.

The escape sequences following *%?2* are only carried out if a parameter value is present; they are ignored if the user does not enter a parameter value for the *%2* variable.

In the preceding example, the *[Disable cluster?]* querey field is defined for a "do if present" conditional variable. Conditional variables can only distinguish whether or not a parameter value is present; they cannot discriminate individual characters in a parameter value. This means that the "do if present" conditional variable is considered true if either **Yes** or **No** is entered in a query field. Therefore, query fields must be left blank to execute the "do if not present" condition.

Testing for Both Conditions

You can group conditional variables together, which tests for both "do if present" and "do if not present," and then execute an escape sequence for the condition that is true.

The following example shows a submit file that executes either with or without user input. If no parameter value is entered for *%0*, the submit file executes without user input; however, if a parameter value is entered for *%0*, the submit file pauses to display a message and prompts the user to press **GO** to continue.

```
%'
Type
%'File list' [Kbd]
%'[Confirm each?]'
●
                Verify that "File.3" was copied.
                    Press GO to continue■
Files
%'[File list(*)]' File.3
%'[Details?]'
%'[Print file?]'
%'[Suppress sort?]'
%'[Max columns]'
%'[Sort by suffix?]'
%'[Suppress error messages?]'
%#0:●:%?0~¢1●●~
```

In the above example, note the following:

1. The %#0 escape sequence specifies that GO (enclosed in colon (:) delimiters) is to be executed if a parameter value is not present for the %0 variable.
2. The **%?0** escape sequence specifies that the *¢1* escape sequence to pause the submit file (enclosed in tilde (~) delimiters) is to be executed if a parameter value is present for the *%0* variable.

Only one escape sequence is carried out, depending on whether or not a parameter value for *%0* is present.

Nesting Conditional Escape Sequences

Conditional escape sequences can be nested, which means that one conditional escape sequence can cause others to be executed. For example, the **OptCmd** command is executed by a submit file. This command is used to optionally execute some other command, which is specified by the user. Whether or not, and subsequently how, the command is executed is determined by nested conditional escape sequences.

The command form for **OptCmd** takes three or more parameter values, as shown below:

```
OptCmd
  Command name   Copy
  [File list]    Vars.fls
  [Parameters]   a1 a2 a3 a4 a5
```

In the *Command name* field, the user enters the name of a command, for example, **Copy**. In the *[File list]*, the user enters the name of a file that contains a list of file names. In the *[Parameters]* field, the user enters the parameter values required by the command specified in the *Command name* field.

When the submit file is executed, the parameter values entered by the user determine what actually takes place.

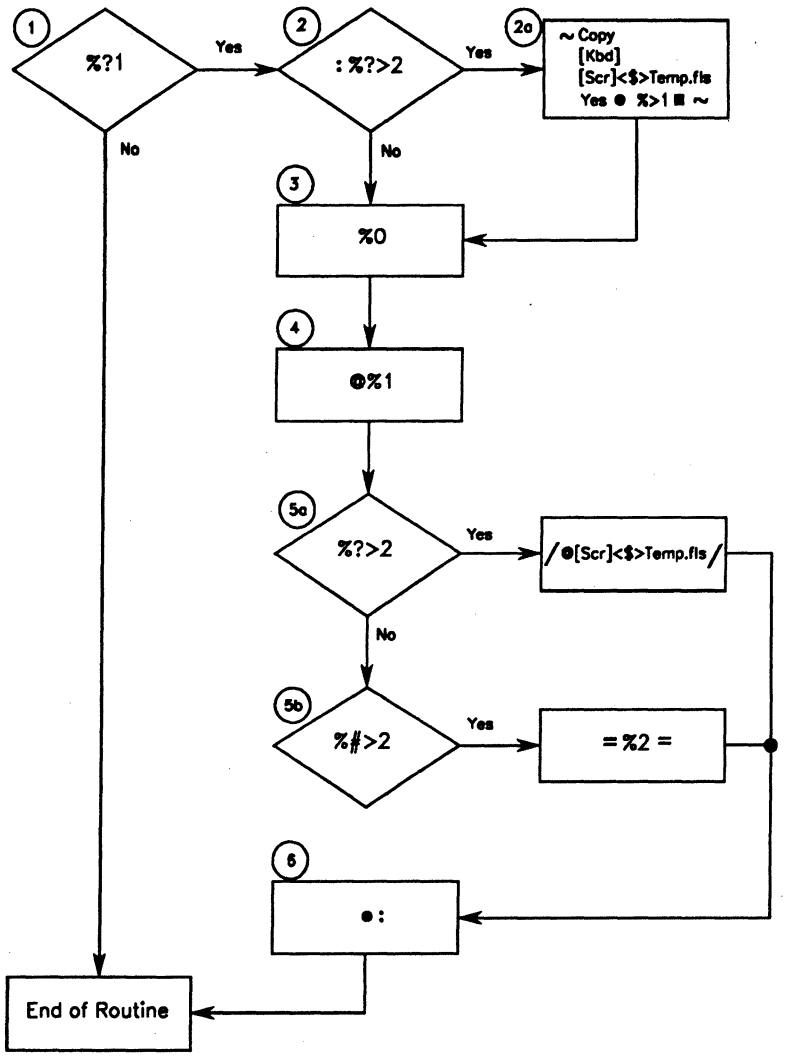
The submit routine for **OptCmd** is shown below.

```
%'
%?1:
%?>2~
Copy
%'File from' [Kbd]
%'File to' [Scr]<$>Temp.fls
%'[Overwrite ok?]' yes
%'[Confirm each?]'
●
%>1■~
%0
@%1
%?>2/@[Scr]<$>Temp.fls/%#>2=%2=
●:
```

Figure 3-20 shows how the escape sequences and commands are carried out when the submit file is executed. Each numbered step is described below.

1. Checks for a parameter value (*%1*) in the *[File list]* field of the command form. If a parameter value is present, goes to Step 2 and executes the commands and escape sequences between colon (:) delimiters. If a parameter value was not entered, skips everything between colons (in this case, everything else in the submit file).
2. Checks for more than one parameter value (*%>2*) in the *[Parameters]* field of the command form. If present, goes to Step 2a and executes commands and escape sequences between tilde (~) delimiters. If not present, skips everything between tildes.
   a. Copies parameter values in the *[Parameters]* field (*%>1*) into a file named *[Scr]<$>Temp.fls*.
3. Expands *%0*. This corresponds to the entry made in the *Command name* field of the command form.
4. Expands *%1* as an at-file. This corresponds to the entry made in the *[File list]* field of the command form.
5. Performs one of the following escape sequences:
   a. If more than one parameter value was entered in the *[Parameters]* field, the file *[Scr]<$>Temp.fls* is expanded as an at-file (see Step 2a, above). This escape sequence is surrounded by slash (/) delimiters.
   b. If only a single parameter value was entered in the *[Parameters]* field, its value (*%2*) is expanded here. This also applies if nothing was entered in the *[Parameters]* field, since a blank value is expanded to null. This escape sequence is surrounded by equal sign (=) delimiters.
6. Executes the command entered in the *Command name* field of the command form (*%0*). This also ends the escape sequence surrounded by colon (:) delimiters, which began in Step 2, above.

Figure 3-20. Submit File With Nested Conditional Variables



541.3-21

Using Nested Submit Files

It is possible for a submit file to execute other submit files and then pick up where it left off. This is called *nesting.*

The example below shows how *SetUp.sub* performs the **Video** command and then submits *DoStuff.sub*. After *DoStuff.sub* has executed, *SetUp.sub* continues to execute, appending a message to a log file.

```
%' This is SetUp.sub
Video
%'[Suppress pause between pages?]' yes
%'[Screen time out in minutes (current)]'
%'[Foreground color (current)]'
%'[Narrow characters]' no
%'[Background color (current)]'
●
Submit
%'File list' DoStuff.sub
%'[Parameters]'
%'[Force expansion?]'
%'[Show expansion?]'
%'[Continue on error?]'
%'[Start after]'
●
Append
%'File list' [Kbd]
%'File to' [Sys]<Sys>DoStuff.log
%'[Confirm each?]'
●
   DoStuff.sub successfully executed.■
```

Nested submit files can also be used with variables. For example, the submit file shown above could be set up as a master submit file to perform preliminary steps, such as the **Video** command. It could then submit another file with parameter values, and finally, write an entry to a log file.

```
%' This is Master.sub
Video
%'[Suppress pause between pages?]' yes
%'[Screen time out in minutes (current)]'
%'[Foreground color (current)]'
%'[Narrow characters]' no
%'[Background color (current)]'
●
Submit
%'File list' %0
%'[Parameters]' %0 %>0
%'[Force expansion?]'
%'[Show expansion?]'
%'[Continue on error?]'
%'[Start after]'
●
Append
%'File list' [Kbd]
%'File to' %0.log
%'[Confirm each?]'
●
   %0 successfully executed.■
```

When *Master.sub* is executed with the **Submit** command, the command form is filled in as follows:

```
Submit
 File list              Master.sub
 [Parameters]           Other.sub d0 Test pswd
 [Force expansion?]     yes
 [Show expansion?]
 [Continue on error?]
 [Start after]
```

The parameter value *Other.sub* is then replaced for *%0* in *Master.sub*, and any other parameter values replace *%>0*.

The nested submit file, *Other.sub*, could look like this:

```
%' This is Other.sub
%' Variables are expanded as follows, from Master.sub
%' %0 is not used, %1 = d0, %2 = Test, %3 = pswd

Path
%'[Volume]' %1
%'[Directory]' %2
%'[Default file prefix]'
%'[Password]' %3
%'[Node]'
●

%' More commands could be executed here.

%' Changes path and password back to where they were
%' set for Master.sub

Path
%'[Volume]' %[
%'[Directory]' %<
%'[Default file prefix]'
%'[Password]' %^
%'[Node]'
●

%' Reverts back to Master.sub.
```

More than one layer of nesting is permitted. For example, another submit file could be executed from *Other.sub*. Nested submit files always revert back to the submit file from which they were called. This is how it works:

Master.sub calls *Other.sub*

Other.sub calls *StillAnother.sub*

StillAnother.sub finishes executing

Other.sub finishes executing

Master.sub finishes executing

Using Recursive Submit Files

It is possible for a submit file to resubmit itself over and over again. This is called *recursion*. Recursion is used to execute a group of commands per file for a number of files. Up to 255 recursions are allowed.

The following example, *CobolTest.sub*, shows how recursion can be used to execute two commands, one at a time, on a list of files. Note that recursion is different from executing one command on a list of files, then executing another command on the same list of files.

```
%' This is CobolTest.sub

COBOL Compile
%'Source file' %0
%'[Intermediate file]'
%'[Listing file]'
%'[For animation?]'
%'[Other directives]'
●
COBOL Run
%'[File name]' %0
%'[Parameters]'
%'[Switches]'
●%?>0:
Copy
%'File from' [Kbd]
%'File to' Temp.fls
%'[Overwrite ok?]' yes
%'[Confirm each?]'
●%>0■
Submit
%'File list' CobolTest.sub
%'[Parameters]' ❷Temp.fls
%'[Force expansion?]'
%'[Show expansion?]'
%'[Continue on error?]'
%'[Start after]'
●:%#>0~¢2~
```

When you execute this submit file, you specify the file names of the programs you want to test in the *[Parameters]* field of the command form, as shown in the following example.

```
Submit
 File list              CobolTest.sub
 [Parameters]           Prg1 Prg2 Prg3 Prg4 Prg5
 [Force expansion?]
 [Show expansion?]
 [Continue on error?]
 [Start after]
```

The submit file then performs the following steps:

1. Executes the **COBOL Compile** and **COBOL Run** commands on *Prg1*, the first file (*%0*) entered in the *[Parameters]* field of the **Submit** command form.

2. Checks for more than one file name in the *[Parameters]* field (*%?>0*). If there, does the following:

   a. Copies all parameters except the first (*%>0*) to a file named *Temp.fls*.

   b. Resubmits itself (*CobolTest.sub*). Notice that by specifying *@Temp.fls* in the *[Parameters]* field, *Temp.fls* is expanded as an at-file. Therefore, all file names copied to *Temp.fls* in Step 2a become parameter values for this recursion of the submit file.

3. Recursion takes place a total of four times. Each time, the COBOL commands at the beginning of submit file are executed on the first file (*%0*) expanded in the *[Parameters]* field.

   During each recursion, the first file name (*%0*), on which the COBOL commands were executed, is dropped from the list of file names copied to *Temp.fls*. Therefore, the parameters list for each iteration of the submit file is decremented by one, as shown below:

First recursion	*Prg2 Prg3 Prg4 Prg5*
Second recursion	*Prg3 Prg4 Prg5*
Third recursion	*Prg4 Prg5*
Fourth recursion	*Prg5*

4. When the conditional variable *%?>0* is expanded to null, steps 2a and 2b are not executed and the submit file is terminated.

Applying Video Attributes

To change video characteristics for an onscreen message, you insert the video escape code, followed by a two character escape sequence, as shown below:

```
⊠AE
```

The following example shows a submit file entry that displays a message in reverse video.

```
%'
Type
%'File list' [Kbd]
%'[Overwrite ok?]'
%'[Confirm each?]'
●
        ⊠AE  This line appears in reverse video.  ⊠AA
■
```

To apply a video attribute, first insert the video escape code. (To do this with the Editor, type **CODE-i ff**, then press **GO**.) Follow it with the appropriate two-character escape sequence for the attribute you want to apply. Video attribute escape sequences are listed in Table 3-6.

Be sure to end all messages with the escape sequence for normal text (AA); otherwise, the remainder of the submit file is executed with the assigned video attribute.

Table 3-6. Video Attribute Escape Sequences

Escape Sequence	Attribute
AA	Normal
AB	Half bright
AC	Underlined
AD	Half bright, underlined
AE	Reverse video
AF	Half bright, reverse video
AG	Underlined reverse video
AH	Half bright, underlined, reverse video
AI	Blinking
AJ	Blinking, half bright
AK	Blinking, underlined
AL	Blinking, underlined, half bright
AM	Blinking, reverse video
AN	Blinking, half bright, reverse video
AO	Blinking, underlined, reverse video
AP	Blinking, underlined, half bright, reverse video

Clearing the Screen for Messages

You can clear the screen before a message is displayed by inserting a special video escape sequence within the message. This makes it easier to notice and read a message, because no command output remains on the screen.

To clear the screen, insert the video escape code, then type **pf**. Follow that escape sequence with the **NEXT PAGE** symbol (a broken line), as shown in the following example:

```
%'
Type
%'File list' [Kbd]
%'[Overwrite ok?]'
%'[Confirm each?]'
●☒PF
— — — — — — — — — — — — — — — — — — — — —
  ☒AE  This message is displayed at the top of the screen. ☒AA

                ☒AE  Press Go to continue.  ☒AA

☒PN■
¢1●
%' More commands would follow here.
```

The PF escape sequence stands for pause off; PN (also shown above) stands for pause on.

Note that a video pause is different from a submit file pause. The video pause escape codes effect output to the screen only; the submit pause escape code (*¢1*) effects execution of the submit file. Therefore, a *¢1* escape sequence must follow the PF escape sequence (as shown above), if you want the submit file to stop executing commands while the message is displayed.

Suppressing the Command Line After Messages

Some users are confused by messages, such as "Press GO to Continue," when the command line is displayed directly beneath it. To suppress the command line, position the **FINISH** symbol for the message after the *¢1* escape sequence, rather than within the text for the message, as shown in the following example:

```
%'
Type
%'File list' [Kbd]
%'[Overwrite ok?]'
%'[Confirm each?]'
●⊠PF
— — — — — — — — — — — — — — — — — — — — —
   ⊠AE  This message is displayed at the top of the screen. ⊠AA

                ⊠AE  Press Go to continue.  ⊠AA

⊠PN
¢1●■
%' More commands would follow here.
```

This delays finishing of the **Type** command in the submit file until the user presses **GO**. There is a drawback, though; if the user types other characters before pressing **GO**, those characters appear on the screen. They do not, however, interfere with execution of the submit file.

In the following example, the **CANCEL** symbol (a bell) is used, rather than **FINISH**. That suppresses the message generated by the **Type** command (that is, "Typed... FileName"), so that no text other than the message appears on the screen.

```
%'
Type
%'File list' [Kbd]
%'[Overwrite ok?]'
%'[Confirm each?]'
●⊠PF
— — — — — — — — — — — — — — — — — — — —

   ⊠AE  This message is displayed at the top of the screen. ⊠AA

                ⊠AE  Press Go to continue.  ⊠AA

⊠PN
¢1●▲
%' More commands would follow here.
```

System Manager

The **System Manager** provides an icon-oriented, mouse-driven command interface. Its display is shown in Figure 3-21. From it, users select function keys to execute commands. The display can be customized to reflect specialized tasks and requirements. See "Configuration File," below, for detailed information.

System Manager has no command form. To start it, type the command name, then press GO.

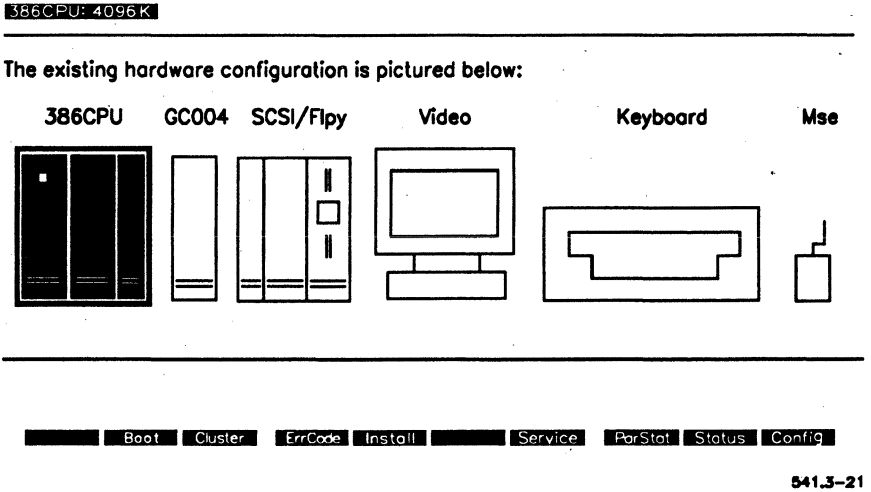
This command is implemented with the run file *SystemMgr.run*, command case 00.

Workstation Display

The **System Manager** displays a picture of the workstation hardware configuration, as shown in Figure 3-21.

To select a module, position the highlight with right and left arrow keys or click it with the mouse. As you move the highlight, information about each module is displayed in the status area.

Figure 3-21. System Manager Display



The display is divided into the following sections:

Status area	The status area is located at the top of the screen, above the double bar. It displays information about the highlighted module and the system date and time.
Hardware components	Workstation modules and other hardware components are displayed and labeled in the center of the screen. The selected module is highlighted (selecting a module is described later in this section).
Function keys	A function key menu is displayed at the bottom of the screen. You use function keys to invoke commands that pertain to the highlighted module. Function key names change as you select different modules.

Function Keys

When you highlight a module, the function key menu displays the commands available for that module. To select a function key, either press the key or click it on the function key menu with the mouse.

After selecting a function key, press **CANCEL** (or click the **CANCEL** button on the mouse) to dismiss a command form or return to a previous menu.

No function keys are available for the keyboard, the Multiline Port Expander, or the PC Emulator.

The following sections describe the System Manager function keys as supplied with Standard Software. If a system has been reconfigured, or if a particular command is not available on the system, the function key selections may differ from those described here.

Workstation Processors

When a workstation processor is highlighted, you can select from the function keys described below. Function keys for SRP processors are described later in this section.

F1 (Remote) — Starts **System Manager** on the server. The **F1** key appears only if a Remote User Manager session is available on the processor to which the workstation's cluster line is connected. If the **F1** key does not appear, start **System Manager** via Cluster View, as described in the *CTOS System Administration Guide.*

F2 (Boot) — Displays the command form for the **Bootstrap** command.

F3 (Cluster) — Displays more function key selections. From this menu, select commands or additional function key sets, as described below:

- **F4** (Status) — **Cluster Status** command
- **F5** (CFA) — Cluster File Access function key set, which includes the following commands:

  CFA Configure, CFA Display Volume Information, Install CFA File Filter, Install CFA Server Service, and **Install CFA Workstation Agent**
- **F8** (Disable) — **Disable Cluster** command
- **F9** (Resume) — **Resume Cluster** command

F4 (ErrCode) — Displays the command form for the **Status Code Reporter** command.

F5 (Install) — Starts the **Installation Manager** command.

F7 (Service) Displays another function key menu. From this menu, select commands or additional function key sets, as described below:

F4 (Net)	**Network Control Center** command
F5 (DCXIns)	Function key set for installing and deinstalling the DataComm Service
F6 (ScrnPrt)	Function key set for installing and deinstalling the Screen Print Service
F7 (Mail)	Function key set for configuring and starting electronic mail
F8 (Cache)	Function key set for invoking the file system cache utilities.
F9 (Stats)	Function key set for installing and deinstalling the Performance Statistics Service
F10 (Others)	Function key for invoking the following function key sets: Math Service Sequential Access Service Queue Manager Spooler Voice/Data Services XBIF Service XC002 Service

F8 (ParStat) Starts the **Partition Status** command.

F9 (Status) — Displays another function key menu. From this menu, select commands or additional function key sets, as described below:

F4 (Clstr)	**Cluster Status** command
F5 (PLog)	**PLog** command
F6 (Version)	**Version** command
F7 (ParStat)	**Partition Status** command
F10 (Printer)	**Print Manager** command

F10 (Config) — Displays another function key menu. From this menu, select commands or additional function key sets, as described below:

F3 (Editor)	Editor application
F5 (CmdFile)	**Command File Editor** command
F6 (CrCnfg)	**Create Configuration File** command
F7 (SetTime)	**Set Time** command
F8 (ListReq)	**List Request Set** command
F9 (MakeReq)	**Make Request Set** command
F10 (Users)	**User File Editor** command

Hard Disks

When a hard disk drive is highlighted, the following selections appear on the function key menu:

F3 (DskSqsh)	Executes the **Disk Squash** command on the selected disk.
F4 (VolStat)	Executes the **Volume Status** command without details, on the selected disk.
F5 (VolDtl)	Executes the **Volume Status** command with details, on the selected disk.
F6 (VrfyVol)	Executes the **Verify Volume** command with details, on the selected disk.
F7 (Files)	Displays the **Files** command form.
F9 (Archive)	Displays more function key selections. From this menu, select commands or additional function key sets, as described below:

F4 (BkupVol)	**Volume Archive** command
F5 (SelBkup)	**Selective Archive** command
F6 (Restore)	**Restore Archive** command
F7 (FmtFlpy)	**Format Disk** command for initializing backup floppies

F10 (DiskUtl)	Displays more function key selections. From this menu, select commands or additional function key sets, as described below:

F2 (Extent)	**FExtent Reporter** command
F3 (VolChB)	**Volume Checkerboard** command
F4 (Format)	**Format Disk** command
F5 (BadSpot)	**Bad Spot Reporter** command
F6 (ChgName)	**Change Volume Name** command

Floppy Disks

When a floppy disk drive is highlighted, the following selections appear on the function key menu:

F4 (VolStat)	Executes the **Volume Status** command.
F5 (Files)	Displays the command form for the **Files** command.
F6 (FlpCpy)	Displays the command form for the **Floppy Copy** command.
F7 (ChgName)	Displays the command form for the **Change Volume Name** command.
F9 (Format)	Displays the command form for the **Format Disk** command.
F10 (FmtBkup)	Executes the **Format Disk** command to initialize floppies suitable for archiving.

Mouse

When a mouse is highlighted, the following selections appear:

F5 (Install)	Installs the Mouse Service.
F6 (DeInst)	Deinstalls the Mouse Service.
F7 (Config)	Starts the **Set Mouse Controls** command.

Monitors

When a monitor is highlighted, the following selections appear:

F4 (SysFont)	Displays the character set and the hexadecimal value of each character (a tool for programmers).
F7 (Info)	Displays the current screen attributes set for the monitor.

Graphics Controllers

Selections for a graphics module or expansion card are the same as those for a monitor.

Ethernet Module

When an Ethernet Module or expansion card is highlighted, the following selection appears on the function key menu:

F7 (Config)	Starts the **Network Control Center** command.

Voice Processors

When a Voice Processor Module is highlighted, the following selections appear on the function key menu:

F4 (Install)	Displays the command form for the **Install Voice Service** command.
F5 (DeInst)	Deinstalls the Voice Service.
F6 (Status)	Starts the **Telephone Status** command.
F7 (Config)	Starts the Operator application.

Tape Module

When a tape module is highlighted, the following selections appear on the function key menu:

F2 (Install)	Displays the command form for the **Install Sequential Access Service** command.
F3 (DeInstl)	Displays the command form for the **Deinstall Sequential Access Service** command.
F4 (Backup)	Displays the command form for the **Volume Archive** command.
F5 (SelBkUp)	Displays the command form for the **Selective Archive** command.
F6 (Restore)	Displays the command form for the **Restore Archive** command.
F7 (TapeCpy)	Displays the command form for the **Tape Copy** command.
F8 (Retensn)	Starts the **Tape Retension** command.
F9 (Erase)	**Tape Erase** command.

Shared Resource Processor Display

If the **F1** (Remote) function key appears on the workstation display, press it to start the **System Manager** on the shared resource processor. In a few moments, the shared resource processor display appears. When a processor board is highlighted, the following function key menu appears:

F1 (Local)	Returns to the workstation display.
F4 (Clstr)	Starts the **Cluster Status** command.
F7 (ParStat)	Starts the **Partition Status** command.

Function keys for disks and tape drives are the same as for workstations. See the complete listing of function keys, earlier in this section.

If your workstation display does not show the F1 (Remote) key, start the **System Manager** via Cluster View. See "Cluster View," in the *CTOS System Administration Guide*.

Configuration File

To customize the **System Manager** function keys, you modify the configuration file, *[Sys]<Sys>SystemMgrConfig.sys*. It consists of internal command definitions, function key definitions, and partition mapping.

Each section is written in following format:

:Keyword:Value

where

:Keyword:	Is recognizable to the **System Manager**, and describes the type of value that follows it. Keywords must be written exactly as shown.
:Value:	Is the variable information you define.

The following paragraphs describe the structure and purpose of each configuration file section. Additional information is available within the configuration file itself.

Internal Command Definitions

The System Manager starts other commands and applications through internal command definitions. Each command definition is written in the following format:

A sample command definition is shown below:

```
:CommandName: FilesNoDetail
:ExecCommand:Files
:Parameter: [~VolumeName~]~PromptUser~
:Parameter:No
:Parameter:~PromptUser~
:Parameter:No
:Parameter:~PromptUser~
:Parameter:~PromptUser~
:Parameter:~PromptUser~
:PauseOnExit: Yes
```

Command definitions consist of the following entries:

:CommandName:

This parameter defines a name for the internal command. You use this name later when assigning the command to a function key.

:ExecCommand: or ***:RunFile:***

This parameter defines an Executive command or run file, respectively.

:Parameter:

This parameter defines values for the specified Executive command or run file. You can specify an explicit value or one defined by the system. System-defined values are described below; they must be surrounded by tildes (~), as shown.

~DeviceName~	Is the device name of the highlighted disk.
~DevicePswd~	Is the password of the highlighted disk.
~VolumeName~	Is the volume name of the highlighted disk.
~ClusterLine~	Is the first cluster line on the selected SRP processor.
~PromptUser~	Displays the corresponding field from the Executive command form, so that the user can enter a value. It is for use in conjunction with the *:ExecCommand:* keyword only.
~ProcessorName~	Is the name of the SRP processor currently selected.

:PauseOnExit:

This parameter is either **Yes** or **No**.

Function Key Definitions

Function keys are defined in groups called *function key sets*, as shown in the following example:

```
:SetName:NgenProcSet
:FunctionKey:F2
:Label:Boot
:CommandOrSet:BootStrap
:FunctionKey:F3
:Label:Cluster
:CommandOrSet:ClusterSet
:FunctionKey:F4
:Label:ErcData
:CommandOrSet:StatusCODERptr
:FunctionKey:F5
:Label:Install
:CommandOrSet:Install
:FunctionKey:F6
:Label:Services
:CommandOrSet:Services
:FunctionKey:F7
:Label:ParStat
:CommandOrSet:~PartitionStatus~
```

A function key set consists of the following keywords and values:

:SetName:

This parameter is predefined by the **System Manager** for each module. Do not change these names.

:FunctionKey:

This parameter defines a particular function key within a set. Its value can be **F1** through **F10**, **FINISH**, **HELP**, **CANCEL**, or **GO**.

:Label:

This parameter defines a character string to appear on the function key menu. It must be seven characters or less.

:CommandOrSet:

This parameter defines an internal command, function key set, or internal function. Internal functions are described below. They must be surrounded by tildes (~), as shown.

~PartitionStatus~	Starts the **Partition Status** command.
~NoStatus~	Indicates that no status is available for a particular icon.
~SystemMgrDisplay~	Displays the **System Manager** screen.
~Exit~	Exits the **System Manager** command.

Partition Mapping

Partition names can be mapped to a function key set. The mapped function key set appears on the **Partition Status** display when the corresponding partition name is highlighted.

The following example shows the configuration file entries for partition mapping.

```
:Mouse Service:MouseSet
```

The keyword surrounded by colons is the partition name, which must be entered exactly as shown by **Partition Status**, including embedded spaces. The value is the name of a function key set.

Tape Copy

Use **Tape Copy** to copy the contents of a file from disk to tape or from tape to disk. You can use this command, for example, to place any single disk file on a tape. You can also use **Tape Copy** to duplicate tapes, which is described later in this section.

This command is implemented with the run file *SeqAccessCopy.run*, command case 00.

Command Form

```
Tape Copy
 File from      _____________________________
 File to        _____________________________
 [Overwrite ok?] _____________________________
```

Parameter Fields

File from

To copy from tape to a disk, specify the device name of the tape drive and the tape file mark from which you want to copy (for example, *[QIC]0*).

To copy a disk file to tape, specify the file specification of the disk file.

File to

When copying to a disk file from tape, enter a file specification.

When copying to tape from a disk, specify the device name of the tape drive to which you want to copy. To append a tape file beyond those that already exist on the tape, specify *[DeviceName]+*.

[Overwrite ok?]

Default: No

This field applies only when you are copying from tape to disk. If you enter **Yes** and the disk file you specified in the *File to* field already exists, the disk file is overwritten. If you leave this field blank, you are prompted for confirmation if the disk file already exists. If you enter **No**, the disk file is not overwritten.

Duplicating Tapes

The following example describes the procedure for making duplicate copies of tapes. Suppose *Tape A* is the original tape, and *Tapes B, C,* and so on, are to be copies.

1. Place *Tape A* in the tape drive.
2. Use **Tape Copy** to copy every tape file on *Tape A* to a disk file. Start with tape file 0, then repeat the **Tape Copy** operation for tape files 1, 2, and so on.
3. Remove *Tape A* from the tape module and replace it with *Tape B*.
4. Use **Tape Copy** to copy the disk files onto *Tape B*.
5. Use **Tape Copy** as many times as necessary to make additional duplicate tapes from the same disk files.

Note that not all tapes are exactly the same length physically. Furthermore, two physically identical tapes are not functionally the same length because of variances in media. Thus, a copy of one completely full tape may require two tapes.

Tape Erase

Use the **Tape Erase** command to erase data on a tape.

This command is implemented with the run file *SeqAccessUtility.run*, command case ER.

Command Form

```
Tape Erase
  [Device name(s) ([QIC])]
  [Suppress user interaction?]
```

Parameter Fields

[Device name(s) ([QIC])]

Default: *[QIC]*

Enter the device name of the drive containing the tape you want to erase. If you specify more than one device, tapes are erased in turn, in the order in which drives are specified.

If more than one drive is specified and an error is encountered (such as no tape in the drive), that error is reported but the command does not exit. For example, if the first tape drive specified does not contain a tape, **Tape Erase** reports that condition, but then proceeds to the next specified drive.

[Suppress user interaction?]

Default: No

By default, **Tape Erase** prompts for confirmation before erasing if the tape contains an archive dataset in the first tape file (*[DeviceName]0*). If you enter **Yes**, the prompt for confirmation is suppressed. If you enter **No** or leave this field blank, you are prompted to press **GO** to confirm erasing the tape.

Note that **Tape Erase** detects an archive dataset in the first tape file only (*[DeviceName]0*). Therefore, if a tape contains some other form of data (such as a file copied to tape with the **Tape Copy** command), the tape erase operation proceeds without confirmation from the user.

Tape Install

Use the **Tape Install** command to install software applications from tape installation media. It can be executed in interactive mode, which is identical to selecting the **Tape Install** option from the **Installation Manager** menu. It can also be executed in noninteractive mode, which eliminates pauses and menus for user input; this mode is generally used in installation scripts.

This command is implemented with the run file *InstallMgr.run*, command case IT.

Command Form

```
Tape Install
  [Non-interactive?]  _____
  [Tape spec]         _____
  [Append to log file?] _____
```

Parameter Field

[Non-interactive?]

Default: No

Enter **Yes** to perform the installation in noninteractive mode. The installation does not pause for user input, and default parameters from the user configuration file are used.

If you enter **No** or leave this field blank, the command runs in interactive mode. You can then choose to display and change installation defaults, as you would with the **Installation Manager** command.

[Tape spec]

Default: *[QIC]0*

Enter the device name and optionally, the tape file mark containing the software you want to install.

[Append to log file?]

Default: No

Enter **Yes** if you want to append to, rather than overwrite, an existing log file. If you enter **No** or leave this field blank, the installation log file is overwritten with each successive installation procedure.

Tape Retension

Use the **Tape Retension** command to retension tapes. Retensioning fast-forwards and rewinds the tape media without affecting data. QIC tapes and half-inch tapes should be retensioned after about two hours of continuous use.

Note: *DDS tape cartridges do not require retensioning. Therefore, the fast-forward operation is not performed on DDS tape media.*

In addition, new tapes, tapes stored for long periods, and tapes exposed to temperatures below 5°C (41°F) or above 45°C (113°F) should be retensioned.

This command is implemented with the run file *SeqAccessUtility.run*, command case RT.

Command Form

```
Tape Retension
 [Device name(s) ([QIC])]
```

Parameter Field

[Device name(s) ([QIC])]

Default: *[QIC]*

Enter the device name of the drive containing the tape you want to retension. If you specify more than one device, tapes are retensioned in turn, in the order in which drives are specified.

If an error (such as no tape in a drive) is encountered, it is reported but the command does not exit. For example, if the first tape drive specified does not contain a tape to retension, **Tape Retension** reports that condition, but then proceeds to the next specified drive.

Telephone Status

Note: *This command is not supported on EISA/ISA workstations.*

Use the **Telephone Status** command to verify the proper operation of *TMService.run* or to debug programs that use the Voice Processor module hardware. See the *CTOS Programming Guide* for a description of the command form and parameter fields.

Command Form

```
Telephone Status
  [Module number]
```

Type

Use the **Type** command to display the contents of a file on the screen.

This command is implemented with the Executive internal value !6.

Command Form

```
Type
 File list
 [Confirm each?]
```

Parameter Fields

File list

Enter the file specification(s) for the file(s) you want to display.

[Confirm each?]

Default: No

Enter **Yes** to be prompted for confirmation before each file is displayed. Press **GO** to confirm, **CANCEL** to deny, or **FINISH** to stop displaying files. If you enter **No** or leave this field blank, all specified files are displayed.

Undelete

Note: *This command is for use with workstation operating systems only.*

Use the **Undelete** command to recover a deleted file. When you delete a file with the **Delete** command, pointers to the data are removed, but the actual data remains on the disk. The **Undelete** command recreates the necessary pointers to the file data. **Undelete** cannot recover files deleted with the **Scrub** command.

The **Undelete** command should be used immediately after an accidental deletion. Chances of recovering a deleted file after any other disk activity takes place are greatly reduced.

This command is implemented with the run file *Undelete.run*.

Command Form

```
Undelete
  [Volume name]          _____
  [Volume password]      _____
  File list              _____
  [Confirm each?]        _____
```

Parameter Fields

[Volume name]

Default: Current path

Enter the volume name of the local disk containing the data you want to recover.

[Volume password]

Default: None

Enter the volume password if the data resides on a password-protected volume.

File list

Enter the directory and file name of the file(s) you want to recover, for example, *<Holstein>MooIDs*. You can specify up to 10 files, however, wild-card characters are not valid in this field.

[Confirm each?]

Default: No

Enter **Yes** to be prompted to press **GO** before each file is recovered.

If you enter **No**, or leave the field blank, you are not prompted to confirm recovery.

Unlock Cache

Use the **Unlock Cache** command to unlock a file that has been locked into the file system cache with the **Lock In Cache** command. **Unlock Cache** also removes the file from the cache.

The file system cache is an area of memory where disk sectors are stored dynamically, as they are used. When sectors are cached, they are accessed from memory, rather than from disk. This increases the speed at which files are retrieved.

For more detailed information, see "Using a File System Cache," in the *CTOS System Administration Guide*. See also the related commands, **Disable Caching**, **Enable Caching**, **Lock In Cache**, and **Cache Status**, in this section.

This command is implemented with the run file *CacheUtility.run*, command case 04.

Command Form

```
Unlock Cache
 File list      _____________________________________
 [Print file]   _____________________________________
```

Parameter Fields

File list

Enter the name of the file(s) to unlock and remove from the cache.

[Print file]

Default: Screen only

If you want a record of screen output, enter a file or printer specification. If the file you specify does not exist, it is created. If it already exists, its contents are overwritten. If you leave this field blank, output is sent to the screen only.

Update Request Set

Use the **Update Request Set** command to do either or both of the following:

- Compare an original and a new request file and report differences between them
- Replace the original request file with the new one

This command is implemented with the run file *UpdateRqSet.run.*

Command Form

```
Update Request Set
 Original request file _____________________
 New request file      _____________________
 [Overwrite ok?]       _____________________
 [Log file]            _____________________
```

Parameter Fields

Original request file

Enter the file specification of the original request file you want to compare with a newer version.

New request file

Enter the file specification of the request file you want to compare the original.

[Overwrite ok?]

Default: No

Enter **Yes** if you want the new request file to overwrite the original one. If you specify **Yes**, the original request file is saved with the suffix *-old* added to the file specification.

If you enter **No** or leave this field blank, the original request file is not overwritten.

[Log file]

Default: None

Enter a file specification or a device to which to write the results of the comparison. If the log file already exists, new output is appended to it. If you leave this field blank, request files are not compared.

The log file is written as follows:

```
:Date:mm/dd/yy

:AddedRequests:
      xxxxh
      yyyyh
:ModifiedRequests:
      zzzzh
:SuppressedRequests:
      bbbbh
```

:AddedRequests: lists requests that are not in the original request file but have been added to the new one. *:ModifiedRequests:* lists requests that have been changed in the new request file. *:SuppressedRequests:* lists requests that are in the original request file but have been removed from the new one.

User File Editor

Use the **User File Editor** command to create and modify user configuration files. With the **User File Editor**, you can add new user names, specify which application is started after SignOn, and modify existing user configuration files.

The menus displayed by the **User File Editor** are defined in a template file, which can be edited to change wording on the menus. Therefore, menus that appear on your screen may be different than those described here. Instructions for editing the template file are included near the end of this section.

This command is implemented with the run file *UserFileEditor.run*.

> **Caution**
>
> Do not use the **User File Editor** to modify a user configuration file that contains optional environments. Doing so deletes the keywords and values for optional environments from that user file. Instead, edit such user files manually; see the *CTOS System Administration Guide* for detailed information about user file environments.

Command Form

```
User File Editor
 User name          _____________________________
 [Old password]     _____________________________
 [New password]     _____________________________
 [Template file]    _____________________________
 [Command file]     _____________________________
```

Parameter Fields

User name

Enter a new or existing user name. User names can be up to 30 characters long and contain any characters except wild cards.

[Old password]

Default: Currently active password

Enter a valid password for the user configuration file you want to edit. If you leave this field blank, the active password (the one you entered at SignOn or set later with the **Path** command) is used.

If the active password or the password you enter in this field is not valid, the following message appears:

```
Access denied  (Error 219)
```

[New password]

Default: No new password assigned

Enter a new password for the user configuration file.

If you specify a new password, which requires a volume or a file password, it is assigned to the user configuration file. You can change this protection level later with the **Set Protection** command, described earlier in this manual.

[Template file]

Default: *[Sys]<Sys>UserFileTemplate.sys*

Enter the name of the template file to be used by **User File Editor**. The template file contains menu items and keyword fields for user file options. (See "Template File," later in this section.)

[Command file]

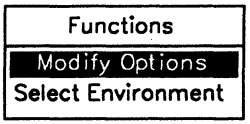
Default: Currently active command file

Enter the file specification of a command file to be used as the default command file for this user name.

Operation

After filling in the command form and pressing **GO**, the **User File Editor** presents a series of progressive menus, from which you select options and edit fields. The Functions menu, as shown in Figure 3-22, is the first menu to be displayed.

Figure 3-22. User File Editor Functions Menu



When a new file is created, a set of defaults, which are defined in the template file, is assigned to the user configuration file. Therefore, even though you may not edit any of the fields, a valid user configuration file can still be created.

While using the **User File Editor**, function key **F2** is operational. **F2** is the undo key, which restores the previously set parameter value to the field you are editing.

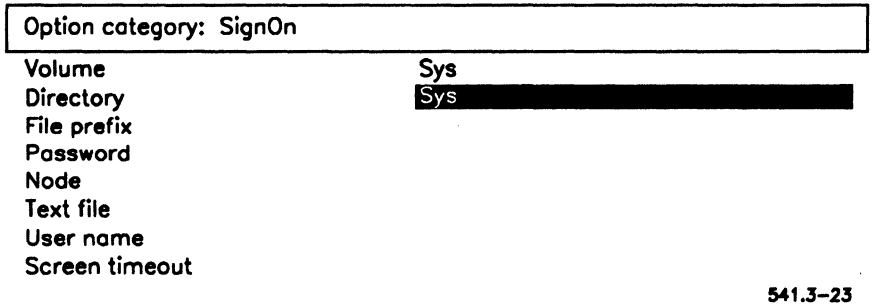
To select from the menu, position the highlight, then press **GO**. To exit the **User File Editor**, press **FINISH**.

Modify Options Menu

The Modify Options menu displays a list of applications and other programs that you can modify with the **User File Editor**. To select a menu item, position the highlight, then press **GO**.

After selecting from the Modify Options menu, an *Options Entry* form appears. An example is shown in Figure 3-23.

Figure 3-23. User File Editor Options Entry Form



The *Options Entry* form includes the name of the program or application, descriptions of user configuration file entries, and the current value for each field. If a field contains a value, you can change it. If a field is left blank, the application assigns a default value.

When you have completed the form, press **GO** to save the changes and return to the Modify Options menu. Press **F2** (Undo) to discard changes; press **CANCEL** to return to the Functions menu without saving changes.

Select Environment Menu

A Select Environment menu is displayed only if the **User File Editor** template file has been customized. From the Select Environment menu, you select the default SignOn user environment. To select an environment, position the highlight, then press **GO**.

See also the *CTOS System Administration Guide* for more detailed information about SignOn environments.

Template File

The menu items and field descriptions displayed by the **User File Editor** are defined within a template file. The template file can be modified to define environment and applications selections.

You can set up more than one template file and specify which one to use when you fill in the **User File Editor** command form (see "Parameter Fields," above).

The default template file, *[Sys]<Sys>UserFileTemplate.sys*, is provided with Standard Software. If no template file is specified when you fill in the command form, the default file is used.

Format

The template file contains the information displayed on the screen by the **User File Editor**. For example, you can change the names of the Select Environments and Modify Options menus by editing the template file. To edit the template file, you use a text editing application, such as the Editor.

Template file entries are written in the following format:

:Keyword:Value

where

:Keyword:	Is recognizable to the **User File Editor**, and describes the type of value that follows it. Keywords must be written exactly as they are shown in "Keyword Descriptions," below. The leading colon must be the first character typed on a line, with the exception of spaces or tabs.
Value	Is the information displayed by the **User File Editor**. Values are the portions of the template file you can change, however some values should not be modified, as described under "Keyword Descriptions," below.

Figure 3-24 shows an example of a user template file. Template file entries are grouped into two divisions. The Options division contains keywords and values for application options. The Environment division contains keywords and values for environments. Indented keywords are hierarchical and must follow the preceding entries, as shown. Keywords and acceptable values are described below.

Figure 3-24. User File Editor Template File

```
:Options:Modify Options
  :Category:Executive
    :Field:ExecCmdFile
      :Description:Command file
  :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

:Environments:Select Environment
  :Name:Executive
    :SignonExitFile:[Sys]<Sys>Exec.run
    :SignonChainFile:[Sys]<Sys>Exec.run
    :Default:
  :Name:Multi Context
    :Command:Install Context Manager
    :SignonExitFile:[Sys]<Sys>Signon.run
  :Name:Mail
    :Command:Mail
    :SignonExitFile:[Sys]<Sys>Exec.run
```

Keyword Descriptions

User template keywords fall into two categories. The first, options keywords, apply to the choices that appear when selections are made from the Modify Options menu in the **User File Editor**. The second, environment keywords, apply to the selections made from the Select Environment menu.

The keywords described below are arranged in logical order, as they would appear within the user template file. Menu items appear on the screen in the order they appear within the template file. See the sample template file, *[Sys]<Sys>UserFileTemplate.sys*, which is supplied with Standard Software, for additional instructions.

Options Keywords

:Options:

The value following *:Options:* appears as a choice on the Functions menu (see Figure 3-22).

:Category:

The *:Category:* keyword is subordinate to *:Options:*. The value following *:Category:* appears as a choice on the Modify Options menu.

:Field:

The *:Field:* keyword is subordinate to *:Category:* and defines an optional user configuration file entry. Do not edit the values for *:Field:* keywords in the user file template. To add new *:Field:* values, check the documentation for the applications you want to include in the user file template.

:Field: values are not displayed by the **User File Editor**. Instead, the value following *:Description:*, which is subordinate to *:Field:*, is displayed as a choice in the *Options Entry* form.

:MustField:

The *:MustField:* keyword is subordinate to *:Category:* and defines a mandatory user configuration file entry. Do not edit the values for *:MustField:* keywords in the user file template. To add new *:MustField:* values, check the documentation for the applications you want to include in the user file template.

:MustField: values are not displayed by the **User File Editor**; instead, the value following *:Description:*, which is subordinate to *:MustField:*, is displayed as a choice in the *Options Entry* form. The value following *:Default:*, which is also subordinate to *:MustField:*, will be added to the user configuration file if no entry is made in the *Options Entry* form.

:FileField:

The *:FileField:* keyword is subordinate to *:Category:* and defines a mandatory user configuration file entry, the value of which must be a file specification. (The file specification is entered later, through an *Options Entry* form in the **User File Editor**.) Do not edit the values for *:FileField:* keywords in the user file template. To add new *:FileField:* values, check the Release Notices and documentation for the applications you want to include in the user file template.

:FileField: values are not displayed by the **User File Editor**; instead, the value following *:Description:*, which is subordinate to *:FileField:*, is displayed as a choice in the *Options Entry* form.

:Description:

The *:Description:* keyword is subordinate to *:Field:*, *:FileField:*, or *:MustField:*. The value following *:Description:* appears as a choice in the *Options Entry* form for the *:Category:* under which it is included in the user file template.

:Default:

The *:Default:* keyword is subordinate to *:Field:*, *:FileField:*, or *:MustField:*. The value following *:Default:* is added to the user configuration file unless it is modified in the Options Entry form of the **User File Editor**.

Environment Keywords

:Environments:

The value following *:Environments:* appears as a choice on the Functions menu (see Figure 3-22).

:Name:

The *:Name:* keyword is subordinate to *:Environments:*. The value following *:Name:* appears as a choice on the Select Environment menu.

:Command:

The *:Command:* keyword is subordinate to *:Name:*. The value that follows *:Command:* must be an Executive command and defines which Executive application is started when the user signs on.

:SignOnExitFile:

The *:SignOnExitFile:* keyword is subordinate to *:Name:*. The value following *:SignOnExitFile:* determines what appears on the screen when the user finishes an application. The value must be a run file and is frequently either *SignOn.run*, which returns the user to the SignOn screen, or *Executive.run*, which starts the Executive.

:SignOnChainFile:

The *:SignOnChainFile:* keyword is subordinate to *:Name:*. It is only used when no *:Command:* value is defined. In such a case, the run file specified as the value for *:SignOnChainFile:* is started when the user signs on.

:Default:

The *:Default:* keyword is subordinate to *:Name:*. It defines the environment that is included in the user configuration file when an environment is not chosen from the Select Environment menu. *:Default:* has no value. It should be included under *:Name:* for one (and only one) of the environment selections.

Verify Volume

Use the **Verify Volume** command to verify file system structures and report the results.

During its first phase, **Verify Volume** checks the integrity of the Volume Home Block (VHB), the File Header Block (FHB), and the master file directory (MFD). It then checks each logical file address (LFA) for disk sectors that overlap or cannot be accounted for. After the verification is complete, **Verify Volume** displays a report of the verification phase.

This command is implemented with the run file *DiskMapper.run*.

Command Form

```
Verify Volume
  [Volume or device name]  _____
  [Volume password]        _____
  [Print file]             _____
  [Summary only?]          _____
```

Parameter Fields

[Volume or device name]

Default: Default volume

Enter the volume or device name of the disk you want to verify.

[Volume password]

Default: Currently active password

Enter the volume password of the disk you want to verify.

[Print file]

Default: Screen only

If you want command output written to a file, in addition to the screen, enter a file specification.

[Summary only?]

Default: No

Enter **Yes**, if you want to display a summary of errors only. If you specify **No** or leave this field blank, a complete listing of files and structures sorted by LFA is displayed.

Display

Figure 3-25 shows the volume verification report displayed by the **Verify Volume** command. Structures (VHB and allocation bit map) and file names are listed in ascending order of their locations on the disk. For every structure or file, a name, beginning and ending address, size in sectors, and allocation error (if any) is reported. Free space on the disk is displayed as "Available."

Figure 3-25. Verify Volume Display

```
Volume Name: Sample                                Mon Jan 21, 1991 6:31 PM

  Name                          Begin       End       Sectors   Errors

<Sys>MasterBootRecord.sys     00000000   00000200         1
<Sys>Crashdump.sys            00000200   00400200      8192
<Sys>BootRecord.sys           00400200   00400400         1
Volume Home Block             00400400   00400600         1
<Sys>SysImage.sys             00400600   00480600      1024
<Sys>Log.sys                  00480600   00486600        48
Available                     00486600   03E74C00    118643
<Sys>                         03E74C00   03E8F600       213
Allocation Table              03E8F600   03E98200        70
Working Volume Home Block     03E98200   03E98400         1
<Sys>Mfd.sys                  03E98400   03EA0E00        69
<Sys>fileheaders.sys          03EA0E00   04CA4E00     28704
Available                     04CA4E00   07930200     91226
<Sys>DiskPartitions.sys       07930200   08B2E000     36847
End of disk                   08B2E000   08B2E000         0

Total Errors: 0      Overlap Errors: 0
```

Version

Use the **Version** command to display the version number and, optionally, more detailed information about run files or libraries. You can also use **Version** to change a run file mode and memory allocations in the run-file header.

This command is implemented with the run file *Version.run*.

Command Form

```
Version
 File list                    _____________________________
 [Details?]                   _____________________________
 [Print file]                 _____________________________
 [Run file mode]              _____________________________
 [Min array, data, code]      _____________________________
 [Max array, data, code]      _____________________________
 [Keyboard ID]                _____________________________
 [Alternate NLS style?]       _____________________________
```

Parameter Fields

File list

Enter the names of run files, libraries, and image files for which you want to display version numbers.

[Details?]

Default: No

If you enter **Yes**, information is displayed about the run-file format and mode, static memory size, minimum and maximum dynamic memory, and character code set.

[Print file]

Default: Screen only

If you want command output written to a log file, enter a file specification. If the file does not exist, it is created.

[Run file mode]

Default: No change

To change the run file mode, enter a valid run file mode. See also the *CTOS Programming Utilities Reference Manual: Building Applications*.

[Min array, data, code]

Default: No change

Enter the minimum number of bytes to place in the run file header for memory array, data, and code, respectively. This entry should consist of three numbers, each separated with a space. See also the *CTOS Programming Utilities Reference Manual: Building Applications*.

[Max array, data, code]

Default: No change

Enter the maximum number of bytes to place in the run file header for memory array, data, and code, respectively. This entry should consist of three numbers, each separated with a space. See also the *CTOS Programming Utilities Reference Manual: Building Applications*.

[Keyboard ID]

Default: None

This field defines a keyboard table other than the system default, to be loaded for the application. Enter a keyboard ID number. See the *CTOS Operating System Concepts Manual* for information about keyboard tables and keyboard IDs.

[Alternate NLS style?]

Default: No

Specify **Yes** to override the default keyboard style the operating system reads from *Nls.sys*. The recognized keyboard style is taken from NLS Table 15, if it is present; otherwise, it is taken from NLS Table 0.

Video

Use the **Video** command to change one or more screen attributes. The changes remain in effect until you log out. Additional attributes can be set with the **Screen Setup** command; see the documentation for it, earlier in this section.

This command is implemented with the Executive internal value !11.

Command Form

```
Video
  [Suppress pause between pages?]                  _____
  [Screen time out in minutes (current)]           _____
  [Foreground color(s) (current)]                  _____
  [Narrow characters?]                             _____
  [Background color (current]                      _____
```

Parameter Fields

[Suppress pause between pages?]

Default: No

Enter **Yes** to suppress the following prompt;

```
Press NEXT PAGE or SCROLL UP to continue
```

This prompt occurs before output is scrolled off the screen. When you do not want to interact with a command, you can suppress the prompt. For example, if you want to leave a tape backup unattended (perhaps overnight), you would suppress the pause between pages.

If you enter **No** or leave this field blank, the Executive prompts you to press **NEXT PAGE** or **SCROLL UP** before information is scrolled off the screen. This is necessary to examine lengthy command output, for example, a long list of files displayed by the **Files** command.

[Screen time out in minutes (current)]

Default: No change

This feature automatically turns off the screen when no characters have been typed within a specified number of minutes. Enter the number of minutes to elapse between the time the keyboard is last used and the screen is turned off.

To turn the screen back on, press any key.

To set the screen to always stay on, enter 0.

Alternatively, this attribute can be specified in the operating system configuration file or in the user configuration file. See "Configuring Workstation Operating Systems" and "Customizing User Environments," respectively, in the *CTOS System Administration Guide*.

[Foreground color(s) (current)]

Default: No change

This field applies to color monitors only. You can set two foreground colors, one for full-intensity and one for half-intensity color, using either of the following methods.

Method 1

For a single foreground color, enter **green**, **white**, **blue**, **yellow**, **amber**, **darkblue**, **purple**, **pink**, **aqua**, **magenta**, **lavender**, **coral**, or **red**.

For two foreground colors, enter the full-intensity color followed by the half-intensity color, separated by space, for example:

```
[Foreground color(s)]  yellow red
```

Method 2

To create your own colors, enter **r** for red, **g** for green, and **b** for blue, followed by a number from **0** to **3** to specify the intensity of each color, for example, **rgb013**. The combination of intensities you enter determines the color you see on the screen.

For two foreground colors, enter the full-intensity color followed by the half-intensity color, separated by space. For example:

```
[Foreground color(s)]  rgb333 rgb013
```

On workstations that support background color, you can enter **rgb000** to produce black characters on the background color you specify (see the *[Background color]* field, below).

See also the *CTOS Editor User's Guide*, for information about setting color with this method, using an interactive display in the Editor application.

[Narrow characters?]

Default: No

This field applies only to monitors having the capacity to display both large and narrow character widths. Enter **Yes** to display narrow characters. If you enter **No** or leave the field blank, large characters are displayed.

Note: *Narrow characters cannot be displayed on SG-2500 workstations.*

[Background color (current)]

Default: No change

On workstations that support background color, select the background color as described above, for the *[Foreground color(s)]* field. To remove background color, which sets the background to black, enter **rgb000**.

Volume Archive

Use the **Volume Archive** command to do the following:

- To back up the contents of a disk to tape, floppy diskettes, or another hard disk
- To back up only those files modified after a specified date (incremental backup)
- To verify the integrity of the volume control structures without backing up the files
- To recover data from a disk that is no longer readable

During a volume backup, users still have access to the disk. When backing up a disk on the server, however, you can disable the cluster to prevent files from being modified and to maximize the accuracy of file system verification. See the **Disable Cluster** command, earlier in this section.

Before you can perform tape backups, the Sequential Access Service must be properly installed. See the **Install Sequential Access Service** command, earlier in this section. See also "Installing System Services" and "Backing Up and Restoring Data" in the *CTOS System Administration Guide*.

This command is implemented with the run file *VolumeArchive.run*, command case VA.

Command Form

```
Volume Archive
 Volume or device name(s)
 [Volume or device password(s)]
 [Incremental from]
 [Suppress backup?]
 [Suppress verification?]
 [Archive dataset ([QIC])]
 [Delete existing archive dataset?]
 [Print file]
 [Display structures?]
 [Verify write?]
 [Suppress user interaction?]
```

Parameter Fields

Volume or device name

Enter the name of the volume or device you want to back up. If you specify more than one, the second and subsequent archive datasets are appended to those that already exist on the tape.

If you are backing up a corrupted volume, you will be prompted to enter the disk code or device type after you press GO to begin the backup.

[Volume or device password]

Default: No password

When backing up a valid volume, enter the volume password. When backing up a corrupted volume, enter the device password.

If you are backing up more than one disk, enter passwords to correspond to the volume or device names you entered above.

[Incremental from]

Default: Date the disk was initialized

Enter a date. Only files that have been created or modified on or after the specified date are backed up. If no date is specified, all files are backed up.

Optionally, specify a time after the date, as shown in the following example:

May 8, 1991 10:00 AM

If no time is specified, all files from 12:00 midnight on the specified date are backed up.

[Suppress backup?]

Default: No

If you enter **Yes**, a backup is not performed, but the integrity of the volume control structures is verified. This is useful when you suspect a problem with a disk.

[Suppress verification?]

Default: No

Enter **Yes** to suppress verification of the file-system integrity after the backup operation.

If you enter **No** or leave this field blank, verification is performed.

[Archive dataset ([QIC])]

Default: *[QIC]*

If you are backing up to floppy diskettes or another hard disk, enter a file specification for the archive files, as shown in the following example:

[Volume]<Dir>d0Archive

A two-digit sequence number, beginning with *.01*, is automatically appended to the archive file name and is incremented by one for each subsequent diskette or hard-disk file in the dataset.

If you are backing up to tape, enter the tape specification, which includes the tape drive name (for example, *[QIC]*) and, optionally, a tape mark specification. Tape drive names are defined when you install the Sequential Access Service; see the **Install Sequential Access Service** command, earlier in this section. The tape mark specification can be either **0**, to specify the beginning of the tape, or **+**, to append data to the end of the tape. If you do not specify a tape mark, the default is 0.

[Delete existing archive dataset?]

Default: No

Enter **Yes** if you want to overwrite the archive dataset that already exists on the backup medium. If you enter **No** or leave the field blank, you are informed if an archive file already exists, and prompted to press **GO** to overwrite it.

[Print file]

Default: Screen only

Enter a file or printer specification, if you want to write command output to a file or send it to a printer. If the file you specify does not exist, it is created. If it already exists, command output is appended to it. If you leave this field blank, output is sent to the screen only.

[Display structures?]

Defaut: No

Enter **Yes** to display a detailed analysis of the volume control structures. This option is most often used only by programmers for file system error analysis.

[Verify write?]

Default: No

This field applies only when you archive data to disk backup media, such as floppy diskettes. It does not apply to tape backups. Enter **Yes** if you want to verify that the data written to the archive file matches the data you are backing up. If data does not match, an I/O error (301) is reported.

If you enter **No** or leave this field blank, verification is not performed.

[Suppress user interaction?]

Default: No

Enter **Yes** to suppress messages that prompt for user interaction, such as mounting the archive medium. When user interaction is suppressed, **Volume Archive** exits with an appropriate error message, rather than pausing for correction of the condition that is preventing it from continuing.

If you enter **No** or leave this field blank, **Volume Archive** will pause and prompt for user interaction, such as mounting the archive medium or putting the tape drive online.

Volume Checkerboard Reporter

Use the **Volume Checkerboard Reporter** command to obtain information about disk fragmentation, in both graphical and statistical form. If less than ten percent of available disk space is contiguous, you should either reinitialize the disk with **Format Disk** or rearrange it with **Disk Squash**. Be sure to back up the disk before reinitializing or squashing a disk.

See also the **FExtent Reporter** command, earlier in this section.

This command is implemented with the run file *DiskUtilities.run*.

Command Form

```
Volume Checkerboard Reporter
  [Device or volume name]  _____________________________
  [Volume password]        _____________________________
  [Print file]             _____________________________
  [Summary only?]          _____________________________
```

Parameter Fields

[Device or volume name]

Default: Current path

Enter the volume or device name of the disk you want to analyze.

[Volume password]

Default: Active password

Enter the volume password.

[Print file]

Default: Screen only

If you want a record of command output, enter a file or printer specification. If you leave this field blank, command output is sent to the screen only. If you specify a printer, the summary information only is printed.

[Summary only?]

Default: No

Enter **Yes** to display a statistical allocation summary only. If you enter **No** or leave the field blank, both the graphical representation and the statistical summary are displayed.

Graphical Display

Figure 3-26 shows an example of the graphical representation of a disk checkerboard pattern. Reverse video indicates that contiguous sectors are in use. Half-bright indicates that some sectors are in use. Normal video indicates that contiguous sectors are available.

Figure 3-26. Volume Checkerboard Display

541.3-26

Statistical Report

Figure 3-27 shows an example of the statistical summary produced by the **Volume Checkerboard Reporter**. A low number of free areas is most desirable, because it indicates the least amount of disk fragmentation. The lowest possible number of free areas depends on the number of bad spots on the disk. You can determine that number by executing **Volume Checkerboard Reporter** immediately after formatting the disk.

Figure 3-27. Volume Checkerboard Statistical Report

```
                              Bytes        Sectors
Total                      20054016          39168
Total In Use                8017408          15659
Total Available            12036608          23509
Largest Contiguous          5989888          11699

Number of Free Areas             35

 512  Bytes/Sector
 512  Bytes/Sector
  16  Sectors/Track
   4  Tracks/Cylinder
 612  Cylinders/Disk
```

Volume Copy

Use the **Volume Copy** command to copy the contents of one volume to another.

This command is implemented with the run file *DiskUtilities.run*, command case 04.

> **Caution**
>
> This command can destroy data. If you specify a destination volume that already contains data, its existing contents are deleted before the volume copy operation begins. If you want to retain existing data on the destination volume, use the **Copy** or **LCopy** command instead.

Command Form

```
Volume Copy
 Source volume or device                ____________________
 [Source password]                      ____________________
 Destination volume or device           ____________________
 [Destination password]                 ____________________
 [Print file]                           ____________________
 [Continue on error?]                   ____________________
```

Parameter Fields

Source volume or device

Enter the volume or device name of the volume you want to copy.

[Source password]

Default: None

Enter the volume password of the source volume.

Destination volume or device

Enter the volume or device name of the volume to which you want to copy. The destination volume must be a local disk, that is, located on the workstation from which the command is issued.

[Destination password]

Default: None

Enter the volume password of the destination volume.

[Print file]

Default: Screen only

If you want a record of command output, enter a file or printer specification.

[Continue on error?]

Default: No

Enter **Yes** if you want **Volume Copy** to continue in case of errors, such as an input/output error (301) on either the source or destination volume or file-in-use errors (220) on the source volume. If you enter **No** or leave this field blank, the command terminates if an error occurs.

Operation

Volume Copy can copy from one type of disk media to another, for example, from 5-1/4 inch to 3-1/2 inch diskettes. However, for it to successfully copy an entire volume, the following conditions must be met:

- The volume to which you are copying (destination volume) must be a valid CTOS volume.
- The disk-space capacity of the destination volume must be large enough to contain the data on the volume from which you are copying (source volume).
- The destination volume's master file directory must be large enough to accommodate all directories from the source volume. (A master file directory, also known as *Mfd.sys*, is created when a disk is initialized.)
- The destination volume must contain enough file headers to accommodate files from the source volume. (The number of file headers is also determined when a disk is initialized.)

Volume Copy creates directories on the destination volume, as they exist on the source volume, before copying files. However, because **Volume Copy** copies to a valid CTOS volume, it does not copy the following system files, which are created during volume initialization:

<Sys>BadBlk.sys
<Sys>BootRecord.sys
<Sys>CrashDump.sys
<Sys>DiskPartition.sys
<Sys>FileHeaders.sys
<Sys>Log.sys
<Sys>MasterBootRecord.sys
<Sys>Mfd.sys

Volume Copy does not change the volume password of the destination volume to match that of the source volume. (Use the **Change Volume Name** command, if necessary, to do that.) In addition, retention of directory and file passwords depends on the password encryption mechanism of the source volume, as described below:

Source Volume	Destination Volume	Result
Unencrypted	Unencrypted	Directory and file passwords are retained
Unencrypted	Encrypted	Directory and file passwords are retained
Encrypted	Unencrypted	Directory and file passwords are stripped before copying to the destination volume
Encrypted	Encrypted	Directory and file passwords are stripped before copying to the destination volume

Volume Status

Use the **Volume Status** command to display the following information about a disk:

- Date it was initialized
- Last date it was modified
- Number of free sectors
- Number of free file headers
- List of directories, protection levels, and approximate maximum files

This command is implemented with the run file *VolumeStatus.run.*

Command Form

```
Volume status
  [Volume or device name]                 _____
  [Details?]                              _____
  [Print file]                            _____
  [Directory spec (*)]                    _____
  [Include temporary directories?]        _____
  [Directory names only?]                 _____
```

Parameter Fields

[Volume or device name]

Default: Default volume

Enter a valid volume or disk device name, with or without square brackets.

[Details?]

Default: No

If you enter **Yes**, detailed information about each directory is displayed in the following fields:

Protection
Maximum Files (approximate)
File Count
Total Sectors

When you enter **Yes** in this field, **Volume Status** forecasts *~Maximum Files* based on the actual number of files currently in the directory and 15 characters per file name for future files. Therefore, the detailed approximation of *~Maximum Files* differs from the one displayed when you leave this field blank, which shows the maximum number of files specified when the directory was created.

[Print file]

Default: Screen only

If you want command output written to a log file, enter a file specification. If the file does not exist, it is created.

[Directory spec (*)]

Default: All directories on the volume

Enter the name of the directory for which you want to display information. This field is useful for displaying details about a specific directory. Wild-card characters are permitted.

To display the **Volume Status** heading (statistical information only, no directory names), enter a pair of single quotation marks (").

[Include temporary directories?]

Default: No

If you enter **Yes**, temporary directories are displayed. Temporary directories are used by the operating system; their names begin with a dollar sign ($), for example, *<$000>*. See *CTOS Executive User's Guide* for more information about temporary directories.

[Directory names only?]

Default: No

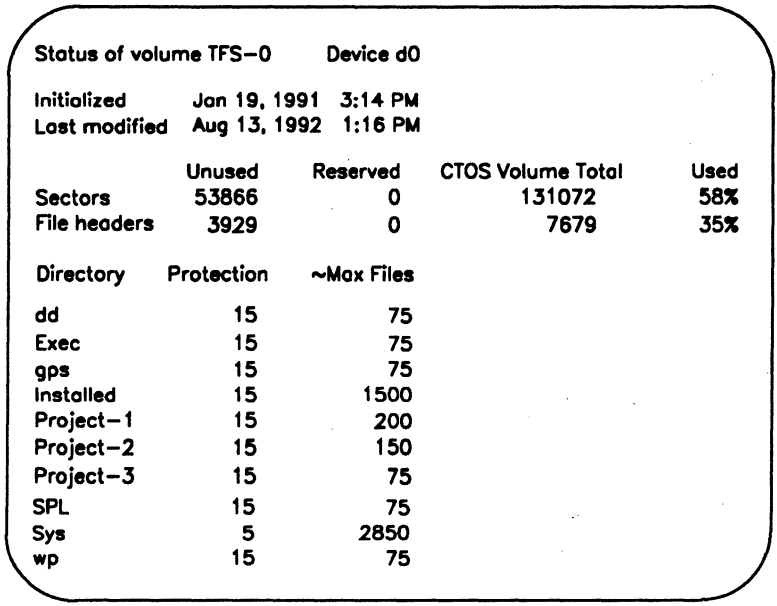
Enter **Yes** if you want to display a list of directory names only, without statistical information. This is useful for quickly checking directory names or creating an at-file. Note, however, that specifying **Yes** in the *[Details?]* field overrides this field and full statistics are displayed.

Display

The **Volume Status** display is pictured in Figure 3-28. The heading shows the volume and device name, date the volume was initialized, date is was last modified, and the following statistics:

Unused	Indicates how many sectors and file headers remain available on the CTOS volume.
Reserved	Indicates how much disk space has been set aside outside of the CTOS partition. (This field pertains to X-Bus+ and EISA/ISA workstations only. See **Format Disk**, earlier in this section.)
CTOS Volume Total	Indicates how many sectors and file headers remain available on the CTOS volume. (If sectors have been reserved for a non-CTOS partition, this field and the reserved disk sectors will equal the total size of the disk.)
Used	Indicates the percentage of sectors and file headers that are currently being used in the CTOS partition.

Figure 3-28. Volume Status Display



```
Status of volume TFS-0        Device d0

Initialized      Jan 19, 1991  3:14 PM
Last modified    Aug 13, 1992  1:16 PM

                 Unused     Reserved    CTOS Volume Total    Used
Sectors          53866             0               131072     58%
File headers      3929             0                 7679     35%

Directory      Protection    ~Max Files
dd                     15            75
Exec                   15            75
gps                    15            75
Installed              15          1500
Project-1              15           200
Project-2              15           150
Project-3              15            75
SPL                    15            75
Sys                     5          2850
wp                     15            75
```

541.3-28

Beneath the **Volume Status** heading, directory names are listed, along with the following information:

Protection — Displays the default protection level of the directory. New files created in the directory will be assigned that protection level.

~Max Files — Indicates the approximate maximum number of files the directory can hold. (The word *approximate* is represented by a tilde (~) on the display.)

Weekly Archive

Use the **Weekly Archive** command to perform a complete volume backup to floppy diskettes, a hard disk, or tape. It combines the following tasks into a single command:

- Cleans up the disk by removing temporary files in the dollar-sign directories
- Performs a complete volume backup
- Records the date and time of the backup in a log file
- Saves the date of the weekly archive procedure, so that you don't have to remember it later when you perform daily backups
- Optionally, disables the cluster if you are backing up the server, and resumes cluster operations when the backup is complete
- Verifies disk volume structures

This command is implemented with the run file *SubCmd.run* and the submit file *WeeklyArchive.sub*.

Command Form

```
Weekly Archive
 Volume to archive      _____________________________
 [Volume password]      _____________________________
 [Disable cluster?]     _____________________________
 Archive dataset        _____________________________
```

Parameter Fields

Volume to archive

Enter the device name of the disk you want to back up.

[Volume password]

Default: Active password

Enter the volume password of the disk you want to back up.

[Disable cluster?]

Default: No

If you are performing a backup of a disk on the server, and you want to disable the cluster, enter **Yes**. If you do not want to disable the cluster, leave this field blank.

Note: *Because* **Weekly Archive** *is executed from a submit file, any response in this field disables the cluster. Therefore, you must leave this field blank if you do not want to disable the cluster.*

[Archive dataset ([QIC])]

Default: *[QIC]*

To back up to tape, enter the device name of the tape drive you want to use. Surround the device name with square brackets, as shown in the command form.

To back up to disk, enter a file specification for the archive dataset, for example, *[f0]<Sys>Archive.*

Method

Weekly Archive first cleans up the disk by deleting all files in the "dollar sign" directories (*<$000>*, *<$001>*, and so on). It then backs up the specified disk and verifies file structures on the disk. Optionally, the cluster is disabled before the backup and resumed when the backup is complete.

A record of the backup is written to *[Sys]<Sys>Archive.log.*

Weekly Archive creates and maintains a file named *[Sys]<Sys>dn>ArchiveTime.txt*, where *dn* is the device name of the disk you are backing up. This file contains the date upon which the weekly archive was performed; it is later used by the **Daily Archive** command to perform incremental backups.

Wrap

Use the **Wrap** command to convert data into an object file.

This command is implemented with the run file *Wrap.run*.

Command Form

```
Wrap
 Data filename
 [Object filename]
 [Module name]
 [Segment name]
 [Public name]
 [Class name]
 [Public?]
```

Parameter Fields

Data filename

Enter the name of the data file whose contents you want to wrap.

[Object filename]

Default: DataFileName.*obj*

Enter the name of the object file where the wrapped data is to be placed.

[Module name]

Default: DataFileName

Enter a name to be used as the internal module name.

[Segment name]

Default: DataFileName

Enter a name to be used as the internal segment name.

[Public name]

Default: DataFileName

Enter a name to be used as the internal public name.

[Class name]

Default: DataFileName

Enter a name to be used as the internal class name.

[Public?]

Default: No

Enter **Yes** if you want the resulting module to have a public segment name.

Example

Use the **Wrap** command to encapsulate data, code, or other programs in an object module format, which can then be linked into an object module by the Linker.

For example, if you are programming in assembly language, you could start your source with the following two statements:

```
segmentname  SEGMENT [PUBLIC] [classname]
             PUBLIC [publicname]
             datafile
```

The names given for the parameters *[Segment name]* and *[Class name]* correspond to the names on the SEGMENT statement. The name you enter in the *[Module name]* parameter is used by the Librarian to refer to the module. The name you enter in the *[Public name]* parameter can be used as the address of the first byte of the data by specifying this name as External in other modules.

Write Hardware ID

Use the **Write Hardware ID** command to assign a workstation hardware identification number or to change the currently assigned ID number. See also "Workstation Hardware IDs," in the *CTOS System Administration Guide*.

This command uses the run file *WHwid.run*.

Command Form

```
Write Hardware ID
 Integer value between 1 and 126 _____________________
```

Parameter Field

Integer value between 1 and 126

Enter a number between 1 and 126 to be used as the workstation hardware ID.

Glossary

A

active password (*also* default password)

The password currently in effect on the workstation.

address

Digital information identifying a location in memory or on a disk.

angle brackets (< >)

Used to enclose a directory name in a file specification or the path setting.

append

To add data to the end of a file that already exists.

application

A program or software package that accomplishes a specific task or set of tasks, such as office publishing, spreadsheet preparation, or database retrieval.

archive dataset

A file or set of files that contains the contents of other files in a compressed form. An archive dataset is created by the **Volume Archive** or **Selective Archive** command.

archive media

Floppy diskettes, hard disks, or tapes that are used to store archive datasets.

at-file

A text file containing a parameter value. At-files are used to supply file lists or other parameter values that are too long to be typed in a command form. When the name of a text file is preceded by the at symbol (@) in a command form field, the contents of the file are substituted as the parameter value when the command is executed.

ASCII text file

A file containing the alphanumeric characters comprising the American Standard Code for Information Interchange. This term is commonly used to denote a file containing unformatted text, as opposed to a file containing imbedded formatting characters, which may not be visible to the user.

asynchronous terminal emulator (ATE)

A workstation functioning as an asynchronous terminal. Also used to describe a software package that provides this capability.

B

bad block table

An area on a disk where the addresses of bad spots are stored. Data is not written to the bad spots identified in the bad block table.

bang (!)

The exclamation point that is typed before the volume or device name to designate a disk on the server.

baud rate

The speed of a data transmission.

binary code

A set of computer instructions reduced to a choice of two alternative conditions.

bit

The smallest unit of data.

BNet

A communications application used to connect servers together. *See also* CT-Net and network node.

bootstrap (*also* boot)

The hardware-initiated process of loading an operating system into memory.

braces ({ })

Used to enclose a network node name in a file specification or path setting.

buffer

An area of memory used as a temporary holding bin for input and output data.

byte

A unit of data containing a specific number of bits.

C

cache

A high-speed buffer that improves data access speed.

case sensitive

Upper case and lower case letters are interpreted differently. The Executive is not case sensitive.

CD ROM

Read-only storage media packaged on compact disk. CD ROM stands for "compact disc read-only media."

channel

A connector to which a peripheral device or communications line can be attached.

circumflex (^)

Precedes a password in a file specification.

click

To press, then immediately release a mouse button.

cluster

A group of workstations connected to a common server. *See also* server.

Cluster File Access (CFA)

A method by which workstations can gain access to disks on other workstations in the cluster.

Cluster View

A software product that connects the keyboard and monitor on a cluster workstation directory to an SRP processor.

cluster workstation

A workstation connected to a server. *See also* server.

code

The contents of a computer program. *See also* source code.

command case

An arbitrary value assigned within a run file that invokes a particular function of the program.

command file

A file the Executive reads to display command names, command forms, and help descriptions, and to associate a run file and command case with each command.

command form

Contains parameter fields for the specified command.

command interpreter

Invokes and passes parameter values to programs.

command line

The highlighted line on the Executive screen where you enter the name of the command you want to issue.

commented text

Text that is ignored when a program is compiled or executed.

conditional variable

An item of data that, depending on its condition, causes a program to execute in predetermined way.

configuration

An arrangement of parts, such as computer hardware, or of elements, such as software programs.

configuration file

Contains parameter values for a software product or hardware device.

Context Manager

A program that divides memory into multiple partitions so that more than one program can be started, and in some cases, simultaneously executed on a workstation.

corrupted volume

An initialized disk that is no longer recognized by the operating system.

crash

See system crash.

crash dump

The process of writing memory to a file, so that it can be examined and debugged by an operating system engineer.

CRC

Acronym for cyclic redundancy check.

CTOS

An umbrella term encompassing all varieties CTOS operating systems. *See also* CTOS I, CTOS II, CTOS III, and CTOS/XE.

CTOS I

Real mode CTOS for workstations.

CTOS II

Protected mode CTOS for workstations.

CTOS III

Protected-mode, virtual memory CTOS for workstations.

CTOS/XE

The family of real mode and protected mode operating systems for shared resource processors.

cursor

A movable marker indicating where the next character typed will appear on the screen.

customized operating system

An operating system that has been modified and rebuilt since the released version. *Compare to* prebuilt operating system.

D

Debugger

A software debugging product used by programmers.

default directory

The directory name that appears in angle brackets (< >) in the path setting on the screen.

default file prefix

The file prefix that appears to the right of <Directory> in the path setting on the screen.

default password (*also* active password)

The password currently in effect on the workstation. Also called active password.

default path

The volume and directory that appear in the path setting on the screen. *See also* path.

default value

A predetermined value with which a command is executed when an optional field is left blank.

default volume

The volume or device name that appears in square brackets ([ ]) in the path setting on the screen.

Development Utility

A command or application that is used primarily by programmers and software development engineers.

device

A disk drive, printer, tape drive, modem, or other physical device that receives or transmits data.

device name

See device specification.

device password

A password assigned to a piece of hardware. Device passwords are assigned by the operating system or in the operating system configuration file.

device specification

The identifier for a piece of hardware. Device specifications are assigned by the operating system or in the operating system configuration file.

device template

A set of configuration file entries that define the physical characteristics of a disk device. Such parameters are required to format a disk.

digital data storage (DDS) drive

A tape drive on a workstation that uses digital audio tapes.

directory

A subdivision of disk storage space.

disk

A mass-storage device for data.

disk drive

The mechanism that holds the disk.

disk drive heads

The mechanisms that read data from and write data to the disk.

disk partition

A subdivision of disk space on X-Bus+ and EISA/ISA workstations, which allows both CTOS and non-CTOS environments to coexist on the same disk.

diskless workstation

Consists of a base unit containing the CPU and video controller. It uses disks on the server for data storage; disks are not connected directly to it.

distribution media

The diskette(s) or tape on which software is supplied.

DMA

Acronym for direct memory access.

dollar-sign directory

A directory that stores temporary files.

DOS

The operating system for IBM PCs.

E

ECC

Acronym for error checking and correction.

Editor

An ASCII text editing application.

EISA/ISA-bus workstation

Consists of a base unit containing a CPU and other hardware components such as a hard disk, a floppy disk drive, and a video controller. It can be upgraded with "industry standard" hardware components. The SuperGen Series 3000 is an EISA/ISA-bus workstation.

error code

A decimal or hexadecimal number denoting an error condition on a workstation or shared resource processor.

escape code

A special-purpose character that causes a deviation from normal program execution.

escape sequence

The deviation of function following an escape code.

expansion

The physical replacement, within a command form, of wild-card characters for the actual characters they match. Also, the physical replacement of the contents of a file when using an at-file as a parameter value.

F

field

The highlighted line in a command form where a parameter value is entered.

file

A set of data that is stored and retrieved as a unit.

file extent

An area of contiguous disk storage space. Although files may appear contiguous when used, a single file often occupies more than one file extent. *Compare to* fragmentation.

file specification

A unique identifier that contains the name of a file, as well as its volume and directory location.

file system

The data and control structures stored on accessible disks.

file system cache

An area of memory on a shared resource processor where file sectors are stored dynamically, as they are used.

floppy diskette

A small removable data storage disk.

floppy disk drive

A slot-like opening on a workstation that holds a floppy diskette.

format

A particular arrangement of data.

format template

A set of configuration file entries that define the characteristics of a volume. Such parameters are required to initialize a disk.

fragmentation

Noncontiguous data storage. A fragmented file is stored in multiple file extents. A fragmented disk contains many small, noncontiguous areas of storage space.

frame

A 4K byte region of physical memory into which the paging service loads a page of program code or data.

function keys

The keys labeled F1 though F10. Their functions change from program to program.

G

Generic Print System (GPS)

A set of software programs that provide printing services for CTOS applications.

H

half-inch tape drive

A reel-to-reel tape drive for use on shared resource processors only.

hexadecimal number

A number in the base sixteen numbering system, which is primarily used by programmers. Hexadecimal digits are represented by numerals 0 to 9 and characters A to F.

histogram

A representation of a frequency distribution.

I

input/output (I/O)

Data transfers between subsystem boundaries, such as from disk to memory, then back to disk.

integrated X-Bus workstation

Contains the processor, a hard disk, and a floppy disk drive in the base unit, and can be expanded with X-Bus modules. The B39 style is an integrated X-Bus workstation. *See also* X-Bus workstation.

J

Job Control Language (JCL)

A programming language processed by the Batch facility.

K

K byte (*also* kilobyte)

1,024 bytes.

keyboard data block

A customizable unit within the system keyboard file.

keyboard table

See system keyboard table.

keyboard literal

The literal character for a keystroke, rather than a function. For example, the keyboard literal for the GO key is •, while its function is to execute a command.

keyword

A predefined word or string that identifies a parameter. Keywords are used in many configuration files and are written in the form of *:Keyword:* colon. The parameter value follows the closing colon.

kilobyte (*also* K byte)

1,024 bytes.

L

library

A collection of object modules that can be linked into run files.

loadable request file

A binary file containing request definitions for a system service.

local file system

A workstation with its own disks, as opposed to a diskless workstation, which always uses disks on the server.

logging out

The opposite of signing on. Logging out exits the Executive.

M

M byte (*also* megabyte)

1,048,576 bytes.

macro

A single keystroke or command that defines a sequence of operations.

magnetic card reader (MCR)

A device used to input data from magnetic cards.

mandatory field

In a command form, a field that must be filled in or the command cannot be executed. A mandatory field is not surrounded by square brackets.

megabyte (*also* M byte)

1,048,576 bytes.

memory

High-speed volatile data storage, the contents of which can be altered at any time. *See also* random access memory.

memory disk

A portion of memory that functions as a disk.

message file

A binary file containing the screen prompts and messages displayed by an application.

message text file

The text source file for a binary message file.

memory partition

A discrete area of memory.

metacharacter

See wild-card character.

modify access

The ability to make changes to a file.

module

A workstation component, such as a disk drive, housed in its own casing and connected as an individual unit.

mouse

An electronic pointing device, used for drawing or selecting items on the screen.

MS-DOS®

The IBM PC compatible operating system developed by Microsoft®.

N

network node

A server connected to other servers via a BNet network. Cluster workstations connected to a node can communicate with other network nodes.

node

See network node.

null device

A valid, but nonexistent device specification *[Nul]*. It is used to test command execution without generating output.

O

optional field

In a command form, a field that provides a default value if left blank. An optional field is surrounded by square brackets ([ ]).

operating system

A program that controls execution of other programs on the computer.

operating system configuration file

A file that contains configurable operating system parameters.

output

Data delivered from a program to the screen, a file, or some other device.

overwrite

To replace the contents of an existing file with the contents of another file. Overwriting destroys the original file. This is an option with many Executive commands.

P

page

A 4K byte section of a program in the linear address space.

paging service

The virtual memory operating system service that controls demand paging.

parameter

A definable element of information affecting the way a program executes.

parameter field

See field.

parameter template

A format or device template read by the **Format Disk** command. *See also* device template and format template.

parameter value

An item of information that is supplied to a program either through a command form or a configuration file.

partition

See Context Manager, disk partition, and memory partition.

password

An access code that restricts use of a system. Workstations or servers can have several passwords that allow varying levels of access to different users.

path

The default volume and directory. This volume and directory are used automatically when you execute a command unless you override the path with a file specification. The path setting appears in the status area of the screen.

prebuilt operating system

A prepackaged, unmodified version of a CTOS operating system that is installed from the distribution media.

primary file headers

The file headers used by the operating system to perform disk read and write operations. *See also* secondary file headers.

primary partition

The memory partition containing the program that is currently active on the workstation or shared resource processor.

processor

The unit that interprets and executes instructions.

protected mode

A program or operating system that can use memory above the first megabyte.

protection level

A number assigned to a file that designates read or modify access and whether a volume, directory, or file level password is required to open it.

pseudovolume (PSV)

A directory on a CTOS volume that simulates a DOS drive.

Q

query field

In a command form, a field that is defined by a field name in the form of a question.

queue

A list of files or jobs waiting for a service, such as printing or batch processing.

Queue Manager

A system service that controls spooled printing and other queue-oriented jobs, such as batch processing.

quarter-inch cartridge (QIC) tape drive

A tape drive on a workstation or shared resource processor that uses cartridge-style tapes.

R

random access memory (RAM)

A high-speed storage area where data is loaded prior to processing. The contents of memory are volatile and can be altered at any time. After processing, data is written back to disk for permanent storage.

real mode

A program or operating system that runs in the first megabyte of memory.

read access

The ability to open or process a file, but not to make changes to it.

record

To store a group of commands that can be reexecuted later.

release documentation

A document containing information about a new version of a software product. Sometimes called Software Release Announcement (SRA), Release Notes, Release Notice, or Release Information File.

replay

To reexecute a group of commands that were recorded earlier. *See also* record.

Remote User Manager (RUM)

A memory manager that allows workstations to execute applications via Cluster View in discrete partitions on a shared resource processor.

request file

See loadable request file.

run file

An executable program.

S

scratch volume

A disk used for storage of temporary files created by some applications.

scroll

To move information on the screen up or down.

SCSI

The acronym for Small Computer Standard Interface. It provides a design standard for hardware device interfaces.

secondary file headers

Duplicate copies of primary file headers. They are used to retrieve data when primary file headers are damaged. *See also* primary file headers.

sector

512 bytes of data.

sequential access

A data access mode in which records are obtained from or placed into a file in such a way that each successive access to the file refers to the next record.

server

A workstation or shared resource processor (SRP) to which cluster workstations are connected. The server controls many system resources, such as printing and communications. Co-workers can share the files and applications that are stored on disks located on the server.

shared resource processor (SRP)

A multiprocessor computer that functions as a server. It is sometimes called an XE.

sign on

The procedure that starts a user session. The user signs on a with predefined name, which determines the applications and commands that are available.

SMD disk drive

An eight-inch hard disk drive for use on a shared resource processor.

Software Release Announcement (SRA)

See release documentation.

source code

The text of a programming language, before it is compiled and linked to form an executable program.

spooled printing

Sends print jobs to a queue on the server, where the Queue Manager assigns priorities and directs documents to the requested printer. Also allows background printing, so that other work can be performed on the system during printing.

square brackets ([ ])

Used to enclose a volume or device name in a file specification or path setting. Also used to designate optional fields in command forms.

squash

To reduce file and disk fragmentation.

SRP

See shared resource processor.

Standard Software

A set of programs, configuration files, and commands that are required to configure the system and perform basic operations.

status area

The top two lines of the Executive screen where the default path, user name, and date/time information are displayed.

status code

A number designating a certain condition on the system. In many cases, status codes represent errors. In other cases, they represent a normal operating condition.

string

A contiguous group of characters.

submit facility

An Executive facility that executes groups of commands.

SuperGen Series 2000 workstation

Consists of a base unit containing the CPU and video controller. It can be upgraded with expansion cards for graphics or extra memory. The SuperGen Series 2000 is a diskless workstation.

SuperGen Series 3000 workstation

Consists of a base unit containing a CPU and other hardware components such as a hard disk, a floppy disk drive, and a video controller. It can be upgraded with "industry standard" hardware components. The SuperGen Series 3000 is also called an EISA/ISA-bus workstation.

SuperGen Series 5000 workstation

Consists of a base unit containing the CPU and removable cartridges, which house disk drives, graphics controllers, tape drives, and other optional equipment. The SuperGen Series 5000 is also called an X-Bus+ workstation.

<Sys>

A directory on every disk that contains unique information about the disk. This information is used by the operating system when you issue commands and use applications.

[Sys]

A system-assigned pseudonym for the volume from which the workstation or shared resource processor bootstraps.

[Sys]<Sys>

The volume and directory containing the bootable operating system.

system crash

An abnormal condition from which the system cannot recover. After a crash, the system usually freezes or reboots automatically.

system error log

A file containing information about many types of hardware and software errors. It can be viewed or printed with the **PLog** command.

system keyboard table

A file named *[Sys]<Sys>NlsKbd.sys*, which is loaded into memory at boot time. It contains keyboard data blocks which define keystrokes for supported keyboards.

system service

A program that expands the capabilities of the operating system.

T

tape file mark

A software mark that separates sequential tape files on a tape.

template file

A text file containing menu displays and function key definitions for a particular command.

TIFF

Acronym for tagged image file format.

U

UNIX

A computer operating system developed by AT&T.

user configuration file (*also* user file)

A configuration containing a user profile.

user name

The name a user signs on with. The file name of a user configuration file, in the form *[Sys]*<*Sys*>UserName.*user*, defines the user name.

utility

A program that carries out a specific task, such as copying or deleting files.

V

valid volume

A disk that has been formatted and initialized for use on a workstation or shared resource processor.

value

An element of information supplied in a command form or a configuration file.

variable

A predefined character or group of characters that is replaced with an actual value during program execution.

variable length parameter block (VLPB)

Used by the Executive to pass parameters to an application run file.

version number

A number designating the revision level of a software product.

virtual memory

The apparent size of memory (to an application), which is greater than actual physical memory size, that makes is possible to address more memory than physically exists on the processor.

volume

An initialized disk. *See also* valid volume.

volume control structures

The framework within which the file system allocates disk space.

volume name

The name assigned to a disk when it is initialized.

W

wild-card character

A special character used by the Executive to match character strings in file specifications. The asterisk (*) wild card matches any number of characters; the question mark (?) wild card matches exactly one character. The tilde (~) is an exclusionary wild card for use with the **Files** command only.

workstation

A desktop computer that can function as a standalone system or be connected into a workgroup called a cluster. *See also* cluster.

workstation-type number (Ws*NNN*)

A three-digit number assigned to the operating system for each type of processor.

write access

The ability to open a file and make changes to it.

X

X-Bus workstation

A collection of separately housed modules, each containing an individual hardware component, such as a processor, a disk, or a graphics controller. X-Bus workstations include the B26, B28, B38, and B39 styles. *See also* integrated X-Bus workstation.

X-Bus+ workstation

Consists of a base unit containing the CPU and removable cartridges, which house disk drives, graphics controllers, tape drives, and other optional equipment. The SuperGen Series 5000 is an X-Bus+ workstation.

XE

See shared resource processor.

Index

D

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

UNISYS

Help Us To Help You

Publication Title

Form Number

Unisys Corporation is interested in your comments and suggestions regarding this manual. We will use them to improve the quality of your Product Information. Please check type of suggestion:

☐ Addition ☐ Deletion ☐ Revision ☐ Error

Comments:

Name Telephone number ( )

Title Company

Address

City State Zip code

Cut along dotted line

Tape

Please Do Not Staple

Tape

Fold Here

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 817 DETROIT, MI

POSTAGE WILL BE PAID BY ADDRESSEE

UNISYS CORPORATION
PRODUCT INFORMATION
MS 18-007
2700 NORTH FIRST STREET
SAN JOSE, CA 95134-2028