

CONTEXT MANAGER MANUAL

Specifications Subject to Change.

Convergent Technologies, Convergent,
CTOS, CT-BUS, CT-DBMS, CT-MAIL, CT-Net, AWS, IWS,
and NGEN are trademarks of Convergent Technologies, Inc.

First Edition (November 1983) A-09-00382-01-A

Copyright © 1983 by Convergent Technologies, Inc.

CONTENTS

GUIDE TO TECHNICAL DOCUMENTATION.....	vii
1 OVERVIEW.....	1-1
WHAT IS THE CONTEXT MANAGER?.....	1-1
Examples.....	1-1
Features.....	1-1
HOW THE CONTEXT MANAGER WORKS.....	1-2
TERMS USED IN THIS MANUAL.....	1-3
STRUCTURE OF THIS MANUAL.....	1-3
2 USER'S GUIDE.....	2-1
EXPERIMENTING WITH CM.....	2-1
AN EXERCISE WITH CM.....	2-2
Starting CM.....	2-4
Using CM.....	2-4
Moving the Highlight.....	2-4
Starting an Application and Assigning a Function Key to It.....	2-4
Overlaying the CM Screen on an Application Screen.....	2-5
Moving the Highlight Between Lists; Returning to the Current Context....	2-5
Starting a Second Application.....	2-5
Switching Contexts Without Using the CM Screen.....	2-5
Eliminating a Context Without Saving	2-6
More Contexts: Swapping Messages...	2-6
Finishing from Within a Context....	2-6
Ending a Session with CM.....	2-6
DETAILS OF CM.....	2-7
Starting an Application and Assigning Function Keys.....	2-7
Function Key Menu.....	2-7
Assigned and Preassigned Function Keys.....	2-8
Starting Applications That Require Parameters.....	2-9
Contexts You Can Return To.....	2-10
Meaning of the Bullet.....	2-10
Status Terms.....	2-10
Swapping a Context to Disk.....	2-11
GETTING OUT.....	2-12
Finishing a Context.....	2-12
Eliminating a Context Without Saving..	2-13
Logging Out.....	2-13

WAYS TO USE CM.....	2-13
Business Tasks.....	2-13
Programming.....	2-14
Communications.....	2-14
3 SYSTEM ADMINISTRATOR'S GUIDE.....	3-1
OVERVIEW.....	3-1
CM and Installed System Services.....	3-2
Installation and Deinstallation.....	3-2
FILES REQUIRED.....	3-2
SWAPPING TO DISK.....	3-3
How Swapping Works.....	3-3
What Contexts Are Swapped?.....	3-4
Time Savings: Swapping	
Versus Starting.....	3-4
The Swap File.....	3-4
PRELIMINARY TASKS.....	3-5
Creating the Swap File.....	3-5
Editing the User Configuration File...	3-6
A Note About the Number of Contexts...	3-7
USING THE CM CONFIGURATION FILE EDITOR..	3-7
Entering from the Executive.....	3-8
Experimenting with the CM Editor.....	3-8
Screen Areas and Functions.....	3-11
Message Line.....	3-11
Input/Error Line.....	3-11
Command Editing Area.....	3-12
Memory Area.....	3-15
Function Key Menu.....	3-16
Command List Area.....	3-18
Exiting the CM Editor.....	3-18
APPLICATIONS YOU CAN START FROM CM.....	3-19
TEXT OF THE CONFIGURATION FILE.....	3-21
EXAMPLES.....	3-22
A Complete New Entry.....	3-22
Editing a Command.....	3-24
Removing a Command.....	3-24
Renaming a Command.....	3-25
A Common Error: Inconsistent Entries.	3-25
4 NOTES FOR THE PROGRAMMER.....	4-1
WHAT RUNS UNDER THE CONTEXT MANAGER?....	4-1
How Video Output Is Handled.....	4-1
Programs that Write Directly	
to the Screen Map.....	4-2
"Busy Wait" Loops.....	4-3
Low-Memory Interrupt Vector Table.....	4-4
Positioning the Cursor.....	4-4
Exit Run File.....	4-5
APPLICATIONS THAT CANNOT BE SWAPPED.....	4-5

COMMUNICATION WITH CM AND BETWEEN APPLICATIONS.....	4-5
ESTIMATING MEMORY REQUIREMENTS.....	4-7
NOTE FOR THE SYSTEM SERVICE WRITER.....	4-8

GLOSSARY.....	Glossary-1
----------------------	-------------------

INDEX.....	Index-1
-------------------	----------------

LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
2-1. Context Manager Screen.....	2-3
3-1. CM Editor Screen.....	3-10

GUIDE TO TECHNICAL DOCUMENTATION

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

Introductory

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

Hardware

NGEN

- Processor Manual: Model CP-001
- Dual Floppy Disk Manual
- Floppy/Hard Disk Manual
- Diagnostics Manual
- Keyboard Manual
- Power System Manual
- Monochrome Monitor Manual: Model VM-001
- Color Monitor Manual

IWS

- Workstation Hardware Manual
- Peripherals Hardware Manual
- IWS Peripherals Hardware Manual (SMD Version)

AWS

- AWS-210 Hardware Manual
- AWS-220, -230, -240 Hardware Manual
- AWS Color Workstation Hardware Manual

Operating System

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

Guest Operating Systems

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

Programming Languages

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

Program Development Tools

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

Data Management Facilities

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

Text Management Facilities

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

Applications Facilities

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

Communications

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

This section outlines the contents of these manuals.

INTRODUCTORY

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

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

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

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

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

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

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

HARDWARE

NGEN

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

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

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

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

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

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

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

IWS

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

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

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

AWS

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

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

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

OPERATING SYSTEM

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

The System Programmer's Guide addresses the needs of the system programmer or system manager for

detailed information on operating system structure and system operation. It describes (1) cluster architecture and operation, (2) procedures for building a customized operating system, and (3) diagnostics.

GUEST OPERATING SYSTEMS

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

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

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

PROGRAMMING LANGUAGES

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

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

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

PROGRAM DEVELOPMENT TOOLS

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

The Editor Manual describes the text editor.

The Debugger Manual describes the Debugger, which is designed for use at the symbolic instruction

level. It can be used in debugging FORTRAN, Pascal, and assembly-language programs. (COBOL and BASIC, in contrast, are more conveniently debugged using special facilities described in their respective manuals.)

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

DATA MANAGEMENT FACILITIES

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

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

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

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

TEXT MANAGEMENT FACILITIES

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

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

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

APPLICATIONS FACILITIES

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

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

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

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

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

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

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

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

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

COMMUNICATIONS

The Asynchronous Terminal Emulator Manual describes the asynchronous terminal emulator.

The 3270 Terminal Emulator Manual describes the 3270 emulator package.

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

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

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

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

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

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

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

MS, GW and XENIX are trademarks of Microsoft Corp.

UNIX is a trademark of Bell Laboratories.

1 OVERVIEW

How many times, while waiting for a task to be completed, have you wished for another workstation so you could get on with another task? The Context Manager gives you the advantages of another workstation by enabling you to run more than one application at once.

WHAT IS THE CONTEXT MANAGER?

The Context Manager is a software product that joins with the CTOS operating system and takes over some of its functions. A workstation with the Context Manager can run several applications at the same time. One of these applications is on the screen; you can see it and work with it. You can switch a different application to the screen with only a keystroke. Then you can return to the first one and continue working at exactly the point where you were before.

EXAMPLES

Suppose you must interrupt your word processing to provide someone with a copy of a file on a floppy disk. You can switch to the Executive, start the Copy command, and return to the same place where you were working in the Word Processor, while the Copy continues to run. You do not have to wait while applications are switched, because each application remains active. The Context Manager (hereafter called CM) does not have to save or open and close files during switching.

CM also allows running more than one of the same application at once. When you are programming, you can write code in one application while another application compiles.

FEATURES

CM has a single screen of its own to which you can return at any time. On the right, this screen lists the applications that you can start. On the left, it shows applications you have started and the status of each one.

Applications may be CTOS-based; or they may run under operating systems hosted by CTOS (for example, MS-DOS or CP/M-86).

CM is not limited to a fixed set of applications. You can configure your system to run whatever applications are important to you.

Because the CM environment is flexible, you can add your own application to it with little or no special programming.

CM saves time and allows you to use more of the power of the workstation than you would if you were running just one application.

HOW THE CONTEXT MANAGER WORKS

Under CM, workstation memory is divided into several partitions. The CTOS operating system resides in the low end of memory. CM itself is in the primary partition. As you start each application, it is placed in one of the remaining partitions. (The person who configures your system determines the total number of partitions. See Section 3, "System Administrator's Guide," for details on configuration.)

Once an application has been started, it is called a context. The most recently started context "owns" the screen and keyboard. When you switch from context A to context B, for example, ownership of the screen and keyboard passes to context B. Other contexts, which can also be running, write whatever they would normally display on the screen to an area in memory.

On some systems, the number of applications that can be started is not limited to the number of partitions. Suppose your system is configured to have three memory partitions available for applications, and you start a fourth application. If the system has been set up to permit it, an application can be swapped to a hard disk (stored temporarily in a suspended state in a disk swap file). CM can handle as many as 10 contexts at once, switching them back and forth from memory to disk as necessary.

TERMS USED IN THIS MANUAL

Some common words have specific meanings as they are used in this manual. They are underlined in the next three paragraphs.

You start an application by choosing it from the list on the CM screen and pressing GO. Once an application is started, it is called a context and is said to be active. You see and interact with one context on the screen. It is called the current context and is said to own the screen.

The context that owns the screen (the current context) is running in foreground. Any other active context is running in background unless it is swapped. An active context is swapped if it has been stored temporarily on a hard disk. While a context is swapped, it is not running, but is suspended.

When you press two keys at the same time, such as ACTION and GO or ACTION and a function key (fn), the combination is symbolized as ACTION-GO or ACTION-fn. When you press certain keys to cause one context to be replaced by another on the screen, you are switching contexts. In this way, you can choose to work in any one of the active contexts by making it the current context.

STRUCTURE OF THIS MANUAL

Before you read this manual, you should be familiar with material in the Operator's Guide (including keyboard layout) and in the manuals for the various applications you plan to run under CM. This manual presumes this knowledge.

For memory and operating system requirements and for instructions for placing CM software on your system, see the Context Manager Release Notice.

The rest of this manual consists of three parts. Each has a different purpose.

Section 2, "User's Guide," explains CM and its operation. If someone else configures your CM, you need to read only this section.

Section 3, "System Administrator's Guide," tells how to prepare for and configure CM. If you are setting up your own system, read Sections 2 and 3. (The term "system administrator," as it is used in this manual, refers to a person who puts new software on a system and configures it. No extensive formal training is implied.)

Section 4, "Notes for the Programmer," contains information for the programmer writing applications to run under CM. If you are writing such programs, you must read this section. You may also find it helpful to read Sections 2 and 3, although it is not crucial to do so.

EXPERIMENTING WITH CM

CM is designed to be self-explanatory. If you are familiar with the workstation already, you can learn your way around CM quickly by experimenting. (If you are a novice, later sections give you more help.) Figure 2-1 shows the CM screen. The boxed summaries on the first few pages of this section tell you what you need to know to use the screen.

You can try out CM quite safely. If you try to do something that is not allowed, a screen message tells you so, but no damage results.

- o The left and right arrow keys move the highlight between "Applications you can start" and "Contexts you can return to".
- o The up and down arrow keys move the highlight within the lists under "Applications you can start" and "Contexts you can return to".
- o The message area tells what you can or cannot do, or explains what has gone wrong.
- o In the CM screen, pressing an assigned function key and then GO starts an application.
- o Pressing ACTION and an assigned function key at the same time switches from one context to another, or from the CM screen to a context.
- o Pressing ACTION-GO overlays the CM screen on the current context screen.
- o CANCEL removes the CM overlay from the current context screen.

SUMMARY OF STATUS TERMS DISPLAYED

Running	This context is running.
Waiting	This context (other than the Executive) is waiting for input.
Done	In the Executive, the last command you invoked has been completed; the Executive is waiting for input.
Swapped	This context has been swapped to disk. It is suspended.
Stopped	This context is in background, but is not running because it is allowed to run only in foreground.

The meanings of these terms are discussed further in "Status Terms," under "Details of CM," below.

AN EXERCISE WITH CM

CM is flexible: it can be set up in many different ways. This section takes you through a series of steps designed to show what your particular CM does.

You must have the CM software to use this exercise effectively. To understand these steps and learn from them, it is best to carry them out actually using CM.

Some details about CM are not covered during this exercise: it is only a general introduction. To learn more, or to get a better picture of CM if it has not been placed on your system, continue on to "Details of CM," below.

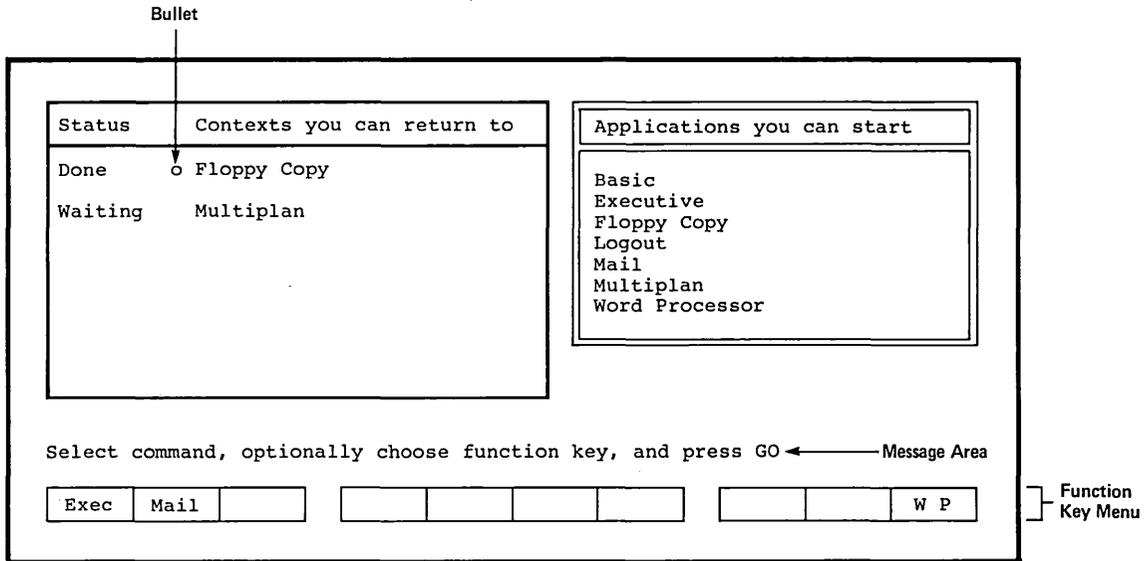


Figure 2-1. Context Manager Screen.

STARTING CM

In most cases, CM is installed automatically when you sign on to your system, and the first display is the CM screen. (The term "install," as it is used here, means "to place in memory, ready to use." For details about installation, see Section 3, "System Administrator's Guide.")

If the first display after you sign on is the Executive command field, type "Install Context Manager", or a unique abbreviation such as "I C M", and press GO. (For details on using the Executive, see the Executive Manual.)

USING CM

The CM screen is made up of four areas that have different purposes. These areas are labeled in Figure 2-1. Refer to the figure and to the boxed summaries in the previous section as you do the following steps.

Moving the Highlight

- o Experiment with the up and down arrow keys to see what they do.

Starting an Application and Assigning a Function Key to It

- o Use the highlight to choose an application to start. Look at the function key menu at the bottom of the CM screen. If a function key already shows an abbreviation corresponding to the application you chose, press GO. If no function key shows such an abbreviation, press any blank function key, and then press GO. Notice what happens on the function key menu.

Any application that has been started is called a context.

Overlaying the CM Screen on an Application Screen

- o Press ACTION and GO at the same time. The CM screen is overlaid on the screen of the application you just started (the current context). Note how the CM screen has changed. (Key combinations are referred to hereafter in the style ACTION-GO.)

Moving the Highlight Between Lists; Returning to the Current Context

- o Note the positions of the highlight and the bullet under "Contexts you can return to." Try out the left and right arrow keys to see what they do, noticing the screen messages as you do so. Now move the highlight to any position on the screen. Press CANCEL.

Starting a Second Application

- o Press ACTION-GO again to see the CM screen. Move the highlight to the "Applications you can start" area; choose another application; assign it to a function key, if necessary; and press GO.

Depending on the kind of system you have, the message area may tell you at this point that a context is being swapped to disk. The same message may occur later on in this exercise; or a message may tell you at some point that swapping does not occur on your system. All these messages are normal. (The concept of swapping is explained under "Swapping a Context to Disk," below.)

Switching Contexts Without Using the CM Screen

- o Press ACTION and, at the same time, the function key assigned to the first application you started (the first context). From now on, this key combination is denoted as ACTION-fn.
- o Press ACTION-fn for the second context.

Eliminating a Context Without Saving

- o Press ACTION-GO. Move the highlight to the first context that you started, and press ACTION-FINISH.

NOTE

When you eliminate a context in the CM screen, your work in this context during this session is discarded. To save work done in a session, you must finish from within each context.

More Contexts: Swapping Messages

- o Start several more (perhaps three or four) applications. With each one, note the messages displayed in the message area. At some point, you should see a message concerning swapping, if you did not see one under "Starting a Second Application," above.

Finishing from Within a Context

- o Press ACTION-GO. Use the highlight to choose an item under "Contexts you can return to" and press GO.
- o With this context on the screen, use whatever method is specified to finish the application involved. If it is the Executive, type "Finish Executive" in the command field and press GO.

Ending a Session with CM

- o Finish all the listed contexts from within themselves, or discard your work in them by choosing them on the CM screen and pressing ACTION-FINISH.
- o Use the highlight to choose Logout under "Applications you can start." Press GO.

This exercise has shown you most of the ways in which you can manipulate applications using CM.

There is more than one way to do some of the things you have done, as you will see in later sections.

You cannot hurt the system by continuing to experiment with CM and learning as you go. If you prefer, you can learn further details by reading "Details of CM," below.

DETAILS OF CM

STARTING AN APPLICATION AND ASSIGNING FUNCTION KEYS

Under CM, each context is assigned to a function key. You use the function keys to switch contexts without having to see the CM screen.

Function Key Menu

Across the top of your keyboard are 10 function keys, f1 through f10. These keys are redefined to do different things at different times and within different applications. The highlighted strip at the bottom of the CM screen is the function key menu, a set of temporary labels for these keys.

On your system, these labels may all be blank, or some may show abbreviations of certain application names. These labels can be configured (set up) differently for each user. (If you are setting up your own system, see Section 3, "System Administrator's Guide," for details on configuration.)

If labels already appear on your function key menu, they have been preassigned for frequently used applications. This preassignment, which is done during configuration, saves time: you do not have to name these function keys every time you use CM.

Assigned and Preassigned Function Keys

When the highlight in the CM screen is within the list of "Applications you can start," the message area says,

Select application, optionally choose function key, then press GO.

Select an application by using the arrow keys to move the highlight to the item you want under "Applications you can start".

A function key may be assigned to an application in any of three ways: it may have been pre-assigned; you can choose a key to assign; or you can let the system assign a key by default.

Your function key menu may show that a function key already has been preassigned to the application you have chosen. After this application has been started, you can use this preassigned function key (in combination with ACTION) to switch to this context from any other context or from the CM screen.

If no function key has been preassigned for the application you are starting, you can choose any of the blank function keys from the menu and assign it to this context. When you follow the screen directions and press the function key you have chosen and then press GO, an abbreviation for the application appears on that key in the menu.

You do not have to choose a function key: you can press GO right away. In any case, when you press GO, the message area says,

Loading . . .

and the first screen of the chosen application appears, replacing the CM screen. Any characters that you type after this message appears go into the type-ahead buffer for the application being started.

If you do not assign a function key at this point, one is assigned for you by default. It is the next available function key, starting from the left on the menu.

If there is a preassigned function key for the application you are starting, you do not have to use it. You can assign another function key to this particular context. This feature is useful when you want to start more than one of an application: two Executives, for example.

A function key label created within a session exists only for that session. When you log out and sign on again, only the preassigned labels are present.

(Note that despite the fact that a function key has been preassigned, you cannot use that function key to enter a context without starting it. You can use ACTION-fn only to reenter an application after it has been started.)

Starting Applications That Require Parameters

Suppose you choose to start an application such as Floppy Copy, to which you want to supply the value of a parameter or parameters. Once you have chosen the application name from the list under "Applications you can start" and pressed GO, the first screen displayed is the appropriate form in the Executive:

Floppy Copy

[Number of Copies]	_____
[Overwrite ok?]	_____
[Dual floppy?]	_____
[Suppress verification?]	_____
[Device name(s)]	_____
[Device password(s)]	_____

After you fill in this form and press GO, the Floppy Copy command is executed.

CONTEXTS YOU CAN RETURN TO

In general, while applications are running, CM responds to the ACTION key. That is why you press this key in combination with other keys to use CM.

To return to a context from the CM screen, you can do any of three things:

- o press ACTION-fn
- o use the arrow keys to move the highlight to the item you want in the list of "Contexts you can return to" and press GO
- o select an assigned function key and then press GO

When you switch from one context to another, you do not finish the first context. It remains active, but moves from foreground to background.

Meaning of the Bullet

Notice the small bullet indicator to the left of one item in the Context list. This bullet shows which context is current. You can return to it simply by pressing CANCEL, even if you have moved the highlight to another place on the screen. (This is useful if you only want to look at the CM screen to check on the status of another context and then go right back to what you were doing.)

Status Terms

The Status column under "Contexts you can return to" may show any of five words: Waiting, Running, Done, Swapped, or Stopped.

A context that is Waiting is doing literally that: waiting for you to type something at the keyboard.

The status Running appears next to a context in which processing is continuing and no input from you is needed at the moment.

The Executive, unlike other applications, tells CM what command has been invoked, and what the status of that command is. If, for example, you give a Files command to the Executive and then press

ACTION-GO to return to the CM screen, you see "Files", and not "Executive", on the list of "Contexts you can return to." When the Files command is complete, its status changes from Running to Done. (Notice that Done does not mean that the Executive itself is finished--only that it has completed the command you asked for.)

The status Done refers only to completed tasks in the Executive. When a long task in any other application is completed, its status is Waiting.

The status Swapped means that CM has swapped that context to disk (stored it there temporarily).

The status Stopped means that this application is in background in a partition, but CM does not allow it to run there. (For example, this arrangement can prevent a program that writes directly to the screen from interfering with CM's normal operation. Such a program can run only in foreground.)

SWAPPING A CONTEXT TO DISK

The memory in your workstation is divided into several partitions. As you start a series of applications, CM places each one in a partition. At some point, you may start one more application than you have room for. If your system has the ability to swap a context to disk, the message area says,

Swapping contexts . . .

There is a short pause, during which CM clears a partition for your latest application by storing another context temporarily on a hard disk. Then the first screen of the new application replaces the CM screen.

Some systems do not have swapping. If yours does not, a screen message tells you so when you exceed the number of available partitions. You must then finish an existing context to make room for a new one.

A context that has been swapped to disk is still a context in the sense that it is still a started application, and you can use ACTION-fn to go back immediately to the same location within it where you were working. But while a context is swapped to disk, it is not actually running. It temporarily suspends its activity until it is called back into a memory partition by CM.

Two kinds of contexts are not swapped: communications applications (such as Asynchronous Terminal Emulator or SNA RJE) and real-time applications (such as direct printing in Word Processor).

CM can keep track of 10 contexts at a time. On most systems, if 10 contexts existed, most of them would be swapped to disk and suspended. Those in memory partitions would be able to run.

GETTING OUT

FINISHING A CONTEXT

If you are working within a context, you can save your work and finish it in the same way as you normally would. When you have given the necessary commands to finish, the CM screen returns, and the message area says,

Finishing . . .

Then the context is removed from the list of "Contexts you can return to."

If you are working in the Executive and want to finish it and return to the Context Manager, type "Finish Executive" (or a unique abbreviation such as "Fin Ex") and press GO.

You can, of course, finish a context that you no longer need at any time during a session.

ELIMINATING A CONTEXT WITHOUT SAVING

If the CM screen is displayed, you can eliminate any context, including the Executive, by moving the highlight to its name on the "Contexts you can return to" list and pressing ACTION-FINISH.

ACTION-FINISH eliminates the context without saving your work from this session. You should use it only for those contexts where there is nothing to save (for example, an Executive with a Done status).

LOGGING OUT

CM does not allow you to log out without finishing or eliminating all contexts.

After you have finished all contexts, you can use the highlight to choose Logout on the list of "Applications you can start" and press GO.

The message area informs you if any context is still active. You must finish all active contexts and then choose Logout again.

If you have the Executive command field on the screen when you decide to log out, and all other contexts are finished, you can log out directly from the Executive by typing "Logout" in the command field and pressing GO.

After you have logged out, the normal SignOn form returns to the screen. Your session with CM is over.

WAYS TO USE CM

BUSINESS TASKS

Suppose you need to read an electronic mail message from your company's finance department, finish up the spreadsheets you are working on in Multiplan, copy them to a floppy disk (using the Executive), write a cover memo (in the Word Processor), and send them out to the address supplied in the finance department memo.

With CM, you can do more than one of these tasks at once: having finished the spreadsheets, you

can start the Copy command in the Executive, and then switch to the Word Processor to write your memo while Copy is still running.

PROGRAMMING

You can write code in the current context while another program compiles in a background context. You can check the status of the background context at any time by pressing ACTION-GO to look at the CM screen. The limitations placed on a background batch application system do not exist with CM. (See "Batch Processing," in the Executive Manual.)

If you need to see display output from a context that is running in background, many applications allow you to pause the output until you choose to examine it in foreground.

COMMUNICATIONS

Suppose you are logged onto an information data base using Asynchronous Terminal Emulator (ATE), or to a mainframe using SNA 3270. You want to check your electronic mail.

You switch contexts, check your mail, and then return to ATE or SNA without ever having to terminate your communications session.

3 SYSTEM ADMINISTRATOR'S GUIDE

OVERVIEW

Configuring CM involves several tasks, which are summarized in the boxed checklist below. Setup is greatly simplified by use of a related program, the CM Configuration File Editor. This section describes the steps you should follow to prepare for and configure CM. You do not have to do these tasks in any particular order.

You can configure CM differently for each user if you want to. Note that you do not need the various application run files in order to do this configuration.

For memory and operating system requirements and instructions for placing the software on the system, see the Context Manager Release Notice.

CHECKLIST FOR THE SYSTEM ADMINISTRATOR

- o Create the swap file or files (if you want the system to be able to swap).
- o Edit the User Configuration file to start CM automatically at SignOn.
- o Use the CM Configuration File Editor to create the CM Configuration file or files.

CM AND INSTALLED SYSTEM SERVICES

CM is not an installed system service. The user can log out of CM and continue to run the system without rebooting.

NOTE

You must install all installed system services before loading CM. Commands such as Install Spooler become illegal after CM has been installed. If you use them, you may have to reboot the workstation.

INSTALLATION AND DEINSTALLATION

For the convenience of the user, CM usually is installed (loaded from disk to memory) from information in the User Configuration file. (It can be left to the user to install CM through a command to the Executive, but this arrangement usually is made only on systems where experimental software is being run outside of CM.)

While applications are running, CM responds to ACTION keystrokes in combination with other keystrokes, and it manipulates applications in response to these commands. (See Section 2, "User's Guide," for a more complete command description. For a general description of how CM works, see Section 1, "Overview.")

CM is deinstalled when the user logs out.

FILES REQUIRED

The following files are required for use by CM. All should be in [sys]<sys>.

CmInstall.run is the run file that installs CM.

Cm.run is the CM run file.

CmNull.run is CM's null exit run file. It is used for communication with CM when an application that was started from CM finishes.

CmConfig.sys (the default) or other file name

is the CM Configuration file. It contains all information used to define names of applications, sizes of partitions, and so on for CM. You may use this default file name or choose any name for a user-specific configuration file. (See "Using the CM Configuration File Editor," below.)

CmConfigEditor.run

is the CM Configuration File Editor run file. This file allows you to edit a CM Configuration file such as CmConfig.sys.

A swap file (name chosen by system administrator)

is the disk file in which CM places contexts swapped to disk. (Further information on creating this file is given under "Preliminary Tasks," below. Swapping is discussed under "Swapping to Disk.")

SWAPPING TO DISK

CM works well whether you set up your system to swap to a hard disk or not. To allow swapping on your system, you must create a swap file, as discussed under "Preliminary Tasks," below.

HOW SWAPPING WORKS

In the process of configuring CM using the CM Configuration File Editor, you identify the memory size of the system and break it down into partitions, or portions of memory, each one of which is interactive and can be occupied by a context. The number of partitions varies with the total memory size, but is often one to three.

As the user starts applications, CM places them in partitions of appropriate size until no more partitions are available. When a further application is started by the user, CM makes a partition available to this new context by temporarily

placing the old context that was in it in a disk file defined for that purpose. This transfer of a context to disk is called swapping.

A context that is in a memory partition is running or waiting for user input in order to run. A context that has been swapped to disk is suspended: it is not actually running. If the user switches to a context that is swapped to disk, CM clears a partition by swapping another context to disk and then swaps the requested context back into the cleared memory partition so that it can run.

Both contexts in partitions and contexts swapped to disk are immediately available to the user without saving and closing and opening of files.

WHAT CONTEXTS ARE SWAPPED?

CM chooses a context to swap based on needed partition size and on the status of all contexts. Also, CM recognizes communications (such as Asynchronous Terminal Emulator or SNA) and real-time applications (such as a Word Processor in direct printing mode) and does not swap them.

TIME SAVINGS: SWAPPING VERSUS STARTING

Part of the benefit of CM to the user lies in the fact that swapping a context from disk to memory is much faster than starting an application. Starting an application often requires opening several files. These files are kept open when a context is swapped to disk.

THE SWAP FILE

CM can swap to a designated file on a local hard disk or over the cluster to the hard disk attached to the master workstation. The swap file should be contiguous (not broken up among different sectors on the disk) so that swapping can be done as rapidly as possible.

Directions for creating the swap file are given under "Preliminary Tasks."

PRELIMINARY TASKS

CREATING THE SWAP FILE

Because the swap file should be contiguous, you should create it immediately after you have initialized the volume (the hard disk) on which it is to reside. For an existing system, this means you should back up, reinitialize, and restore the volume just before creating this file. (See discussions of the IVolume command and of Backup Volume and Restore in the Executive Manual.)

To create the swap file, use the Create File command in the Executive as follows:

Create File

File name	<u>filename</u>
[Volume or Directory password]	_____
[File password]	_____
[File protection level (default = 15)]	_____
[Size in sectors (default = 0)]	<u>nnn</u>
[Overwrite ok?]	_____

To swap to a local hard disk, use the format [volname]<dirname>filename to designate the file. To swap over the cluster, designate the file as [!volname]<dirname>filename.

Use a unique swap file name for each cluster workstation. A name can identify the user or the workstation: for example,

```
[sys]<sys>EricksSwapFile
```

To specify the size of the swap file in sectors (shown as nnn in the example above), you must guess how many contexts might be swapped to disk during a busy session. The size of the swap file is at least the sum of the sizes of the applications that are swapped.

The size of an application usually is given in K bytes; the size of the file must be specified in sectors. The relationship between the two is

2 sectors = 1K bytes

Thus, a 200K-byte application occupies 400 sectors. A swap file large enough for five 200K-byte applications contains 1000K bytes (1M bytes) or 2000 sectors.

The largest swap file this user could ever need theoretically would contain a number of sectors equal to two times the size of the largest application being used times 10 (the maximum number of contexts). Although you may want to calculate this number to find out what the maximum would be, you should not specify this size for the swap file. A pragmatic size is much smaller.

So that CM can swap applications to the correct disk file, you identify this swap file to CM by using the Swap (f8) function key in the CM Configuration File Editor. (See "Function Key Menu," under "Using the CM Configuration File Editor," below.)

EDITING THE USER CONFIGURATION FILE

Most users prefer to have CM installed automatically after they have signed on. If this installation is to be automatic, each User Configuration file should conform to this example:

```
:SignOnExitFile:[sys]<sys>SignOn.run
:SignOnChainFile:[sys]<sys>CmInstall.run
'Install Context Manager'
[sys]<sys>filename
```

where the file name is the default (CmConfig.sys) or the file name you define for this user when you use the CM Configuration File Editor. (See "Using the CM Configuration File Editor," below.)

You can edit the User Configuration file directly, or you can use the User Configuration File Editor to do so. (See the Executive Manual.)

Note that this type of installation allows CM to run in an environment that does not have the Executive.

If you prefer not to have CM installed automatically, the user must install it from the Executive command field by giving the command Install Context Manager (or a unique abbreviation such as I C M) and pressing GO.

A NOTE ABOUT THE NUMBER OF CONTEXTS

It is unusual to have more than three partitions on a system. Should you create a great many more, however, you may at some point receive this screen message:

This version of the OS cannot support any more contexts.

In this situation, refer to the Context Manager Release Notice for a discussion of system build.

USING THE CM CONFIGURATION FILE EDITOR

The CM Configuration File Editor is not part of CM. It is a separate program that allows you to supply information about how each system is to be set up. This information includes the size of the workstation memory and the number of memory partitions that you want, as well as details about the applications that you want a particular CM to be able to run.

All this information is placed in a CM Configuration file. When CM is installed, it refers to this file for specific details on how it is expected to function.

The CM Configuration file is not the same as the User Configuration file. The User Configuration file is more general in scope. (See "User Configuration," under "Advanced Concepts," in the Executive Manual.)

ENTERING FROM THE EXECUTIVE

To start the CM Configuration File Editor from the Executive, use the command CM Configuration File Editor, or a unique abbreviation.

CM Configuration File Editor

[Configuration file name] _____

It is preferable to enter the name of the configuration file at this point, although you can provide it within the CM Configuration File Editor (hereafter called the CM Editor). The default file name is [sys]<sys>CmConfig.sys.

If not all users are to have the same configuration, use the CM Editor several times to define the various configurations you want, and give distinct names to these configuration files.

Each user's own User Configuration file must contain the name of that user's CM Configuration file. Enter the CM Configuration file name when editing the User Configuration file. (See "Editing the User Configuration File," above.)

EXPERIMENTING WITH THE CM EDITOR

As with CM itself, it is not difficult to learn your way around the CM Editor by experimentation. The software is well protected; and a message tells you if you try something that is not allowed.

After you use the CM Configuration File Editor command, fill in the configuration file name, and press GO, the CM Editor screen appears. All the necessary editing is done in this single screen. Its main areas are labeled in Figure 3-1.

In general, RETURN, NEXT, or the up and down arrow keys move the highlight from field to field within a given area. The left and right arrow keys move the edit cursor within a field. The usual editing commands (DELETE, CODE-DELETE, OVERTYPE, and so on) work within a field. (See the Forms Manual for details.)

The CM Editor has one function key menu. Its commands are listed in the boxed summary below.

A step-by-step example of an editing session is given under "Examples," below. Details about each area and field of the CM Editor screen appear under "Screen Areas and Functions," below.

CM EDITOR FUNCTION KEY MENU

Memory (f1)	Moves the highlight to the first field of the memory editing area
Undo (f2)	Replaces the value in the current field with the immediately previous value
Show (f3)	Replaces the CM Editor function key menu display with a display of the current values on the CM screen function key menu
Check (f4)	Carries out verifications of the information supplied
Create (f5)	Records the command name entered in the Command Name field as a new command
Remove (f6)	Removes the name in the Command Name field from the command list
Rename (f7)	Allows renaming of the command entered in the Command Name field
Swap (f8)	Allows identification of the swap file corresponding to this configuration file

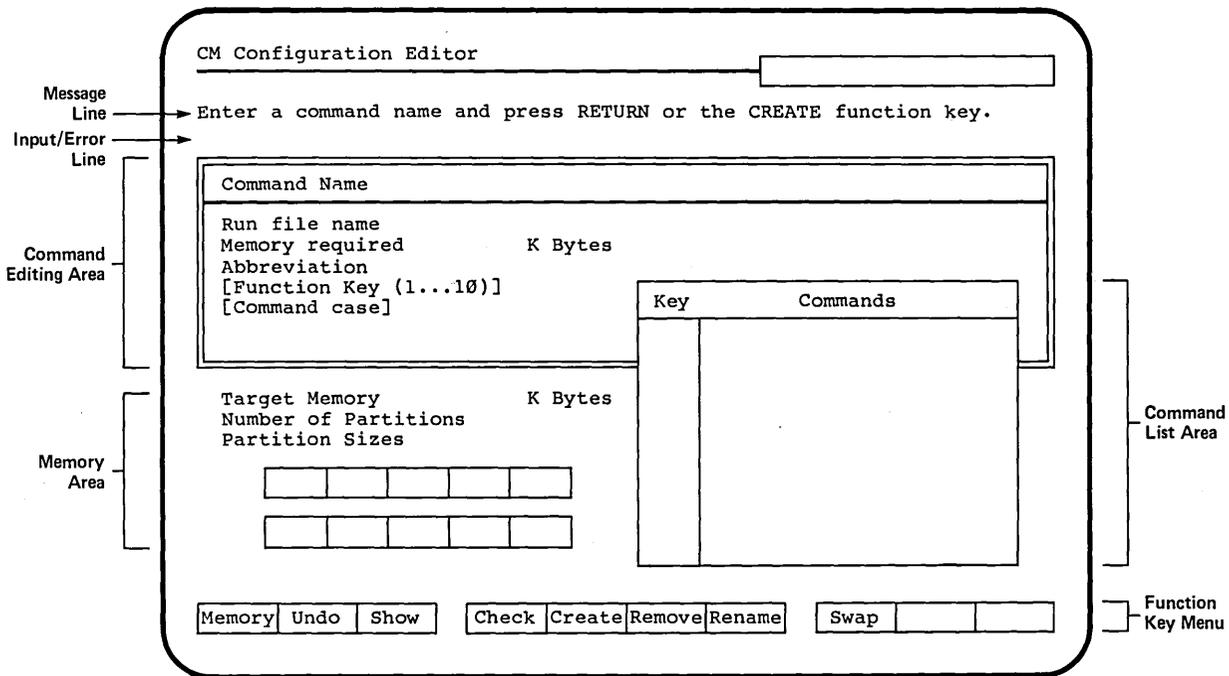


Figure 3-1. CM Editor Screen.

SCREEN AREAS AND FUNCTIONS

This section describes in detail the functions of each screen area. It is not a sequential tutorial. To see what a step-by-step session with the CM Editor is like, refer to "Examples," at the end of this section.

Message Line

The first line below the status line of the editor screen is the message line. During most of your work, it tells you what you can or should do next.

When the CM Editor screen first appears, if you provided a CM Configuration file name in the Executive command field, the message line says,

```
Enter command name and press RETURN or the
CREATE function key.
```

If you did not already provide a CM Configuration file name, the message line says,

```
Please enter a configuration file name.
```

Input/Error Line

The next line of the CM Editor screen is the input/error line. If the message line asks for a configuration file name, the input/error line is highlighted because it serves as the input area for this file name. The default file name [sys]<sys>CmConfig.sys appears in this highlighted line. To accept this default name, press GO. To define a new file name, edit within this line as you wish, and then press GO.

If you enter a CM Configuration file name that does not exist and press GO, the message line informs you and asks for confirmation that you want to create this file. Press GO to confirm that you do, or CANCEL to discard it.

Once you have chosen the CM Configuration file name, the input/error line has the main function of telling you when and why you cannot do

something that you have tried. Meanwhile, the message line tells you what to do next.

The input/error line also tells you the results of the Check operation, if there are results you should know about. (See "Function Key Menu," below.)

The messages that appear in these two lines are self-explanatory.

Command Editing Area

You can add an application (such as Word Processor or Multiplan) or an Executive command (such as Floppy Copy or Files) to the list that CM recognizes. No matter to which category your choice belongs, it is called a command in the CM Editor screen, because that is its function. Once it appears in the list on the CM screen, however, it is called an application.

The large rectangular box surrounded by a double rule is the command editing area. (See Figure 3-1.) You can use it to name and describe a command or application that you want to add to the list of applications that the user can start. You also use this area to edit or remove commands that already are listed.

In the CM Editor, you do not have to edit fields in a certain order, and you do not have to complete all fields in one session.

If you press CANCEL when the highlight is on any of the command editing area fields, the current entry in the Command Name field and all the edits in the command area fields are discarded, and the highlight returns to the Command Name field.

Note that you must define the command Logout, with the run file [sys]<sys>SignOn.run, as one of the allowed commands. Otherwise, the user cannot log out from CM.

Command Name Field. In this field, enter the name of the command or application you wish to add, edit, or remove. Type the name of a new command exactly as you want it to appear in the list of "Applications you can start" on the CM screen (for example, Word Processor).

A name of 32 characters can be displayed both on the command list in this screen and on the list of "Applications you can start" in the CM screen. (You can define a name of up to 50 characters; the lists show only the first 32 characters.)

You can press either RETURN or Create (f5) at this point. If you are adding a new command, press Create (f5).

If you are editing information for an existing command, press RETURN. If the name you have entered is not identical to one on the list, the command is created. (Note that if you are editing a command rather than defining a new one, you can enter only as much of the command name as is needed to make it unique.)

If you are renaming a command without changing any other information about it, or if you are removing a command from the list, you can press Rename (f7) or Remove (f6) without pressing RETURN first.

If you press RETURN or Create (f5) after filling in the Command Name field, the highlight that was on this field moves to the next field, the Run File Name field.

If you enter a new command name in the Command Name field and press GO, the new name is listed immediately in the command list area.

Run File Name. In this field, you can enter or edit the name of the run file for the application or command you entered in the Command Name field. You do not have to have this run file present on your system to do this configuration.

Do not delete or edit the run file name [sys]<sys>SignOn.run for the Logout command.

Note that you can press GO at this point and enter a new command name in the list without filling in this Run File Name field. This flexibility allows you to do a partial entry or edit. The CM Editor assumes that you will eventually reenter this field and supply the run file name before trying to use the command in CM.

In the command editing area, RETURN, NEXT, or the up or down arrow keys cycle the highlight within this area only. (To reach the Memory area, press Memory (f1).)

For a discussion of run file names for applications that require parameters or assume an Executive screen, see "Applications You Can Start from CM," below.

Memory Required. Enter the amount of memory required by this application. (See the Context Manager Release Notice for memory requirements for standard applications; see also Section 4, "Notes for the Programmer," for estimation of memory requirements for other applications.)

Abbreviation. This abbreviated name for the command appears on the function key menu in the CM screen if you or the user assigns a function key to this application.

The default value is the first six characters of the Command Name field entry. You can edit this field.

[Function Key (1...10)]. You can choose to pre-assign a function key to this application so that the user does not have to do so at each session. (For a description of how these labeled function keys are used, see Section 2, "User's Guide.")

You can see which function keys are already preassigned by looking at the "Key" column at the left of the command list area. Enter the number of any available function key.

You can press Show (f3) to display the preassigned function keys on your own function key menu. In this way, you can check the arrangement of function keys to see if it is convenient for the user. Press CANCEL to dismiss the function key menu displayed by Show (f3).

You do not have to make any entry in this field.

NOTE

Do not assign all 10 function keys to applications. At least one should be left blank so that the user can start applications that do not have preassigned function keys.

[**Command Case**]. Some applications (notably the Word Processor) carry out different processes depending on what command was used to invoke them. (For the Word Processor, these commands are Word Processor and Recover.) The command case is a two-character parameter that is passed to the application to tell it which command has been given, and thus what process should be done. (See the section on parameter management in the CTOS Operating System Manual for a further discussion of command case. See also "Adding a New Command," under "Advanced Concepts," in the Executive Manual.)

If the application you are describing can be invoked by more than one command, you must enter a value for the command case parameter in this field. For the correct parameter value, see the release notice for that application.

Some applications, when started, require parameters or assume that they are being called from the Executive. These applications require the entry CM in the Command Case field. (For details, see "Applications You Can Start from CM," below.)

The default value of Command Case is 00.

Memory Area

The memory area is labeled in Figure 3-1. To enter this area at any time during an editing session, press Memory (fl). The highlight goes to the Target Memory field.

If you press CANCEL at any time while you are in the memory area, the highlight returns to the field where it was when you pressed Memory (fl).

Target Memory. Enter the size of the memory of the workstation for which this configuration is being done. A default value is shown in this field. You can edit it to the desired value.

Number of Partitions. You determine (and enter here as one or two digits) the number of partitions into which to divide the workstation memory. (The default value is 1.) This number is governed by the total memory size and by the sizes of the applications that you want to be able to run. The maximum number of partitions is 10, but a more practical number usually is one, two, or three.

Partition Sizes. At the bottom of the memory area are two rows of five highlighted blocks each. One of these blocks is allowed for each possible partition. (A default value is given for the first block.)

Enter the size of each partition (in K bytes). Up to four digits are allowed. Judge the sizes of the partitions according to the applications you want to run. A good working partition size is 250K bytes. The minimum size is 15K bytes.

The left and right arrow keys move the edit cursor within a block in this field. NEXT, RETURN, or the up and down arrow keys move the highlight from field to field within the memory area. You are allowed to make entries in only the number of blocks corresponding to the number of partitions that you specified.

Press GO to record your changes and exit from the memory area, or CANCEL to exit without recording changes. In either case, the highlight returns to the field where it was when you pressed Memory (fl).

Function Key Menu

For a general discussion of the use of the function keys in CM, see Section 2, "User's Guide."

The boxed summary at the beginning of this section briefly describes the effect of each function key in the CM Editor.

At any given time while you are working in the CM Editor, only those labels that are appropriate at that time are shown on the menu.

Most of the commands in this menu have been discussed in earlier subsections. Some require further information.

Undo (f2). This command replaces the value in the current field (the field indicated by the highlight) with the immediately previous value in that field. In numeric fields, if the immediately previous value was blank, Undo (f2) replaces the current value with 0.

Check (f4). This command carries out verifications, comparing some items of information you have supplied to others. It reports any discrepancies in the input/error line.

Remove (f6). Remove (f6) allows deletion of a command name from the Command Name list. The highlight can be on any field of the command editing area (including the Command Name field) when you press this key.

A screen message asks you to confirm deletion of the command. Press GO to confirm deletion, or CANCEL to retain the command on the list.

Rename (f7). Rename (f7) allows you to change the name of a command. Enter the old name of the command in the Command Name field. You can press Rename (f7) immediately, without pressing RETURN.

Edit the field to show the new name of this command. When you press GO, the highlight moves to the Run File Name field. You are not, however, required to edit the run file name.

Press GO to record this change of name, or CANCEL to discard it.

Swap (f8). This command allows you to identify the swap file name you created to correspond to this configuration file. (See "Creating the Swap File," under "Preliminary Tasks," above.) When you press Swap (f8), the message line instructs you to enter a swap file name. Do so in the input/error line, which is highlighted, and press GO to record it or CANCEL to discard it.

Command List Area

The boxed list of commands with their assigned function key numbers, shown in Figure 3-1, is the command list area. This list shows the currently allowed commands: that is, those recognized by CM under this configuration file. (On the CM screen, these commands are called "Applications you can start.")

This screen area is not an editing area, but reflects what has been recorded from other editing areas in this session and in previous ones.

CM can recognize 18 commands that you define through the CM Editor. CM can manage a maximum of 10 contexts, or started applications, at once. It is important to understand this difference of meaning: you are not restricted to defining 10 commands.

Note that if you have entered the CM Editor from an Executive running under CM in order to edit your own configuration file, the changes you make do not appear on the "Applications you can start" list in the CM screen until you log out and then reinstall CM, at which time CM reads the CM Configuration file again.

Exiting the CM Editor

When you have completed your editing session, press FINISH to exit the CM Editor. A screen message asks you to confirm that you want to finish. The input/error line tells you if any inconsistencies have been found. You have the opportunity to remain in the CM Editor and correct them.

If you want to overwrite the configuration file with your edits, press GO. Note that it is only at this point that any of your edits become part of the configuration file, even though some of your changes have already appeared in the command list area in the screen.

If you decide not to save your changes, press CANCEL.

APPLICATIONS YOU CAN START FROM CM

Applications that can start directly from CM must follow two rules:

- o They must be able to be invoked without parameters.
- o They must begin by clearing the screen before writing anything.

Examples of such applications include BASIC, the Executive, Mail, and Multiplan. When you are using the CM Editor to define these commands, you can specify their associated run files in the Run File Name field.

Applications that do not follow both of these rules still can be invoked indirectly from CM. They are invoked through the Executive.

As an example, suppose you want to be able to invoke Floppy Copy from CM. Floppy copy has optional parameters that are passed to it through the Executive. Even more important, Floppy Copy does not clear the screen: it writes to the screen at the next available location.

To add Floppy Copy to the list of applications you can start, use the CM Editor as follows.

- o In the Command Name field, enter "Floppy Copy".
- o In the Run File Name field, enter [sys]<sys>Exec.run (instead of [sys]<sys>FloppyCopy.run).
- o Enter the appropriate memory size, abbreviation, and optional preassigned function key.
- o In the Command Case field, enter the two characters "CM".

Note that to define the Executive itself as a command, rather than Floppy Copy, you should leave the Command Case field blank.

Under this arrangement, when you select Floppy Copy from the list of applications you can start in CM, the Executive is actually invoked. The Executive follows both rules: it can be invoked without parameters; and it clears the screen before writing.

Part of the Executive's purpose is to supply a convenient way to pass parameters to an application. When the Executive begins execution, it checks to see what case value was used when it was invoked. The case value "CM" indicates that it was invoked from CM. The name of the command you specified (Floppy Copy) is passed to the Executive, which then brings up the appropriate form. You fill out the form and press GO to start Floppy Copy. When it is complete, it returns to CM.

Some applications do not recognize command case values, but assume that they have been called from the Executive and derive the equivalent information from the number of lines in the command form. These applications also should be invoked through the Executive and defined with "CM" as the command case value. If you define one of these applications directly, the following message appears when you try to start it in CM:

This application must be invoked through the Exec; edit the Config File.

No damage results from this error. Use the CM Editor to specify [sys]<sys>Exec.run as the run file and CM as the command case for such an application.

TEXT OF THE CONFIGURATION FILE

You can use the Type command in the Executive to look at the CM Configuration file itself.

This example of a file contains one command (the Executive). A series of entries like that shown here for the Executive should exist for each command you have described using the CM Editor.

This file has been created by the Configuration File Editor for use by the Context Manager. The file consists of fields that are recognized by :FieldName:, followed by a parameter. Fields must begin in column 1.

Note:

TargetMemory and MemorySize are specified in K bytes.

Command descriptions are assumed to be in sorted order.

Blanks are significant after the colon.

```
:SwapFile:[sys]<sys>cmSwapFile.sys
:TargetMemory:768
:NumberOfPartitions:2
:PartitionSize:220
:PartitionSize:220
```

```
:CommandName:Executive
:RunFileName:[sys]<sys>Exec.run
:CommandAbbreviation: Exec
:MemorySize:200
:CommandCase:
:FunctionKey:1
```

} For
Each
Command

EXAMPLES

A COMPLETE NEW ENTRY

1. In the Executive, use the CM Configuration File Editor command and enter the configuration file name. A file name for a specific user might be of the form

cmUserConfig.sys

where "User" within the file name can identify a person, if you find this helpful.

Press GO.

2. The CM Editor screen appears with the highlight on the Command Name field. You want this user to be able to start the Word Processor. Type in the name "Word Processor" (not including the quotation marks) and press Create (f5).
3. The highlight moves to the Run File Name field. Type in the name of the Word Processor run file. Press RETURN.
4. In the Memory Required field, enter the amount of memory required for the Word Processor. (See the Context Manager Release Notice.) Press RETURN.
5. In the Abbreviation field, the default entry "Word P" has been clipped from your entry in the Command Name field. Suppose you feel that "W P" would be a better abbreviation.

Press CODE-DELETE to remove the current entry. Type "W P" (not including quotation marks). Press RETURN.
6. You decide to assign function key f1 to the Word Processor. In the Function Key field, type "1". Press RETURN.
7. The Word Processor is an application that does require a Command Case field entry. See the Word Processor Release Notice for your version of the Word Processor to get the correct two-character entry.

8. Press GO to record your entries for this command. (If you press RETURN instead, the highlight goes back to the Run File Name field.) The name "Word Processor" and the function key name "1" appear in the command list area, and your entries for this particular command are complete.
9. Because this configuration file is a new one, you must provide memory and partition information. To enter the memory area, press Memory (fl). The highlight goes to the Target Memory field.

Use CODE-DELETE to delete the default value. Enter the amount of memory of the workstation. Suppose it is 768K bytes: enter "768". Press RETURN.

10. In the Number of Partitions field, delete the default value and enter "2", which is a good working number of partitions for this user's needs. (The applications to be used require medium to large partition sizes.) Press RETURN.
11. Since you chose two as the number of partitions, the CM Editor allows you to fill in partition sizes in only the first two blocks in the Partition Sizes field.

Suppose you decide by looking at the applications that you intend to place on this system that one partition should be of 250K bytes and the other of 300K bytes.

Make these entries in either order, deleting the default value in the first block. Press RETURN or NEXT to get from the first block to the second. (The CM Editor does not let you move to the third block or beyond.)

12. When you have provided correct information in the memory area, press GO.

You can now continue to add commands to the list by typing in another command name and filling in the necessary information. You do not need to reenter the memory area unless you decide to change the information within it.

13. When you have completed all your work on this configuration file, press FINISH. Press GO to confirm recording the entries you have made in this file.

EDITING A COMMAND

Suppose that when you were entering new commands in this user's configuration file, you added Multiplan and assigned it to function key f6. The user now asks you to change the function key abbreviation from M P to Multi.

When entering the CM Editor, you give the name of this user's configuration file in the CM Configuration File Editor command form:

```
cmUserConfig.sys
```

Press GO. The highlight goes to the Command Name field. Enter the command name "Multiplan" and press GO. When the highlight enters the command editing area, use RETURN, NEXT, or the down arrow key to move it to the Abbreviation field. Press CODE-DELETE to remove "M P", and type in "Multi". Press GO.

If you have no other changes to make, press FINISH and confirm overwriting this user's file by pressing GO.

REMOVING A COMMAND

Suppose that when you first defined this user's configuration file, you added the Copy command to the list of available commands because you thought Copy would be used often. This turns out not to be true, and the user wants it removed.

As before, you enter the CM Editor, specifying cmUserConfig.sys as the configuration file name.

In the Command Name field, enter the name of the command you want to remove, "Copy". You can now press Remove (f6), and then press GO in response to a message that asks you to confirm the deletion. The name is removed from the list. If you have no other changes, press FINISH and GO to overwrite the file.

RENAMING A COMMAND

On this user's command list, you originally entered "Asynchronous Terminal Emulator" as a command name. The user would like the name to read "ATE".

After entering the CM Editor, you enter in the Command Name field the old name, "Asynchronous Terminal Emulator" (or only as much of that name as you need to distinguish it from other commands).

You can now press Rename (f7), delete the old name, and type in the new name: "ATE". When you now press GO, the highlight moves to the Run File Name field, but you do not have to edit this field, even though you are given the opportunity to do so. (In this case, the run file does not change.) Press GO again to record your change. The new name replaces the old one on the list.

A COMMON ERROR: INCONSISTENT ENTRIES

Suppose that you are making some changes in a user's configuration file. You accidentally type in "800" instead of "200" as the memory required for one application, and you press RETURN without noticing your error.

The input/error line notifies you of the error (your memory requirement exceeds the target memory of the system) and your workstation beeps. You are allowed to continue the edit, however, because the CM Editor allows for the fact that you may intend to enter the memory area and change information there.

If you leave this error in place and try to finish the session, however, you are reminded of it, and you have the opportunity to remain in the CM Editor and change it.

Where an error cannot be compensated for by an entry in another field, the current field does not allow you to exit. For example, if you specify function key "20" in the Function Key field, you are not allowed to exit that field, because no other entry can make "20" correct.

4 NOTES FOR THE PROGRAMMER

From the application programmer's viewpoint, the Context Manager makes itself part of the CTOS operating system and takes over certain of its functions.

WHAT RUNS UNDER THE CONTEXT MANAGER?

With the Context Manager installed, most existing applications can run without change, and in fact without recompilation or relinking.

Programs that are affected are the following:

- o a program that writes directly to the screen map in memory
- o a program that contains a "busy wait" loop that does not allow the processor to give up the keyboard when the program is waiting to read keystrokes
- o a program that changes a value in the low-memory interrupt vector table directly in memory
- o a program that positions the cursor using the video controller port
- o a program that changes the exit run file

In addition, it is possible to run only one graphics program at a time under the Context Manager, because graphics output does not go to the character map, but instead goes to the graphics board.

Similarly, communications ports cannot be shared by two programs. For example, you cannot run two versions of the Asynchronous Terminal Emulator at the same time, because both want to use channel A.

HOW VIDEO OUTPUT IS HANDLED

Each memory partition has a system structure called the video pointer map; and each has an application character map. The video pointer map is an array of pointers, one for each line of the

screen, that always point to the location of the associated line of the application's screen. The application character map is an array where lines of the screen are stored when an application is running in background.

When the user switches from context A to context B, three things happen:

- o The real screen is copied into application A's application character map.
- o Application B's application character map is copied to the real screen.
- o Each pointer in both video pointer maps is updated to point to the appropriate position either on the real screen or in an application character map. It is these individual pointers that allow the Context Manager's own screen to appear to overlay the current context's screen when the user presses ACTION-GO.

PROGRAMS THAT WRITE DIRECTLY TO THE SCREEN MAP

The mapping of video lines insures that the output from all write requests goes to the correct place: either to the real screen or to the application character map. The Video Access Method (VAM) and Video Display Management (VDM) do this mapping invisibly to the application. Therefore, programs that use VAM and VDM calls work automatically. Programs that write directly to the screen, however, cannot be detected; and the output from their write requests goes directly to the screen whether they own the lines of the screen or not.

The 9.0 and later versions of the CTOS operating system provide three System Common procedures that you can use to allow your programs to work:

LockVideo

UnLockVideo

GetpStructure

For details on the use of these procedures, see the CTOS Operating System Manual.

During initialization, you must first get a pointer to the video pointer map. (The name of the video pointer map is "rgpVidMemLine".) You do this by calling GetpStructure and passing a value of 7 (lGetpRgpVidMemLine). The returned pointer points to the video pointer map.

Later in the program, when you want to move characters to the screen, you must lock all the video structures so that they will not be changed by other programs while you are using them. This is done by a call to LockVideo. To do the actual write, you need to have a structure that looks like a single line of the video and is based on the appropriate pointer entry in the video pointer map. When your write is finished, you must unlock the structures by calling UnLockVideo. The series of locking, writing, and unlocking should be done on a line-by-line basis.

Because these procedures lock the video during a write, they insure that one program that wishes to write to the screen does not interfere with another one that is writing at the same time.

NOTE

If these calls are not used, CM does not permit the program to run in background: it suspends the program and shows the user the status "Stopped".

"BUSY WAIT" LOOPS

In dealing with the keyboard, a program should not use a "busy wait" loop.

The normal way to read keystrokes from the keyboard is to issue a CTOS ReadKbd. A busy wait may be generated by a program that uses calls to ReadKbdDirect with mode 1. Since mode 1 means that the operating system returns immediately whether a character is available or not, the program might be written to loop, waiting for a character to arrive.

When a program is running in foreground, the Context Manager raises its priority (That is, gives it a lower numeric value); when a program runs in background, its priority goes back to normal. If a program contains a busy loop and is running in foreground, its priority is higher than those of other programs in background. Because this foreground program never waits, background programs can never get any processor time, and therefore they stop running.

LOW-MEMORY INTERRUPT VECTOR TABLE

If a single application is running, it may change any value in the low-memory interrupt vector table without problems. Because more than one application can run under the Context Manager, each application must tell the operating system that it wants to change one of the values in this table. To do this, use either of the following CTOS calls:

SetIntHandler

SetTrapHandler

Use SetIntHandler if your application should not be swapped: for example, if the handler that you are providing is a hardware interrupt service routine.

Use SetTrapHandler if you want to allow your application to be swapped out. For example, if the handler that you are providing is a software interrupt service routine, your program can be swapped, because no interrupts of this type can occur while the program is swapped out. (Refer to the CTOS Operating System Manual for more information.)

POSITIONING THE CURSOR

Since a program cannot know whether it is running in foreground or background, it must not output values to the video controller port to move the cursor. Instead, the CTOS procedure PosFrameCursor should be used to position the cursor.

EXIT RUN FILE

When the user selects an application to start from CM, CM chains to the run file and sets the exit run file to CmNull.run. Therefore, when the selected application finishes, CmNull.run runs. CmNull.run sends a message to CM asking CM to terminate this context. (See "Communication with CM and Between Applications," below.) Upon receiving this message, CM terminates the context that sent it.

If you write an application that changes the exit run file, then CM never gets the message that the program has ended. Therefore, in such a case, you must provide a way to run CmNull.run.

For example, the Executive sets itself as the exit run file. Thus, when an application that was started from the Executive finishes, the Executive is reloaded. To allow the user to get back to CM from the Executive, a new command, Finish Executive, has been added. This command simply invokes CmNull.run.

APPLICATIONS THAT CANNOT BE SWAPPED

Certain calls to CTOS, such as SetCommIsr and SetIntHandler, identify an application as one that cannot be swapped. Real-time and communications applications that include such calls are recognized as not swappable, and CM never tries to swap them to the disk.

If you wish to insure that your application is not swapped out, add a call to the CTOS routine SetSwapDisable. From there on, the application cannot be swapped out. (See the CTOS Operating System Manual for more information.)

COMMUNICATION WITH CM AND BETWEEN APPLICATIONS

Applications can communicate with CM using the Interprocess Communication facility (IPC) in CTOS. (See the CTOS Operating System Manual for more information on the use of IPC.) CM sits at an exchange waiting for the user to type some ACTION-keystroke or for a message from an application. An example of the latter is when the Executive sends a message to CM telling it the last command that the user entered.

The communication is done via an operating system call, `NotifyCM`, whose definition is as follows:

`NotifyCM(msgType, pbMsg, cbMsg): ErcType`

where

`msgType` is a word value code for one of the following messages:

- 1 Terminate this context and pass an `erc` (sent by `CmNull`)
- 2 New command (sent by the Executive)
- 3 Logout (sent by `SignOn`)
- 4 Terminate this context if you wish (sent by the Executive)
- 5 Install the CM (sent by `CmInstall`)
- 6 Graphics application (sent automatically by all graphics applications)
- 7 Terminate this context and pass an error message (sent by `CmNull`)
- 8 This context can run in background (clean) (sent by any application)
- 9 This context cannot run in background (dirty) (sent by any application)

`pbMsg/cbMsg` describes the appropriate message. These are only meaningful to CM for messages of types 1, 2, 4, and 7. Values 8 and 9 deserve further explanation. Depending on the sequence of calls that a program

makes to video routines, CM marks that context as clean or dirty (able or unable to run in background). Any program can override this marking by including an explicit call to NotifyCM with the values 8 or 9.

Applications that are running in separate partitions under CM can communicate with each other in any agreed-upon manner. They can use their own definition of messages using the IPC facility.

ESTIMATING MEMORY REQUIREMENTS

The memory requirement of each defined application must be supplied to the CM Configuration File Editor when the system is configured.

For an application that you write, typically you can obtain the run file size from the map file generated by the Linker. Find the last entry in the second column of the map file: this hexadecimal entry is the size in bytes. In the example below, the correct entry is shown in boldface.

<u>Start</u>	<u>Stop</u>	<u>Length</u>	<u>Name</u>	<u>Class</u>
00000H	0016CH	016DH	EXBUF_CODE	CODE
0016EH	00516H	03A9H	EXCLOK_CODE	CODE
.
:	:	:	:	:
.
075C0H	07D0FH	0750H	STACK	STACK
07D10H	07D10H	0000H	MEMORY	MEMORY
07D10H	07D10H	0000H	??SEG	

Convert this value to decimal and add the constant 10240 (the 10K partition overhead).

Examine all calls to AllocMemoryLL and AllocMemorySL, find the total of all memory that the program could allocate while running, and add this total to the above value.

Divide the result by 1024, rounding up if there is a fractional remainder.

If your program has overlays, the procedure is the same except that you should take from the map file the last entry in the second column before the "Overlay 0" subheading.

In the case of a program that tunes itself based on the amount of memory available (that is, calls AllocAllMemorySL), the situation is less well defined. The best approach is to instruct the system administrator to create as large a partition as is practical to accommodate this run file, and then to identify a run file size to fit this partition.

NOTE FOR THE SYSTEM SERVICE WRITER

On systems that run CM, system services should handle swapping requests as described in the CTOS Operating System Manual.

GLOSSARY

Active. An application that has been started under CM (a context) is said to be active.

Application. An application is a program with which a user can carry out tasks such as word processing or financial spreadsheet operations. An application is not a context until it has been started by CM.

Application character map. The application character map is an array where lines of the screen are stored when an application is running in background.

Background. Any context other than the current context is in background.

Bullet. On the CM screen, the bullet is the small dot indicator to the left of the name of the current context under "Contexts you can return to."

CM Configuration file. The CM Configuration file contains information required by the system to run CM for a given user.

Command case. The command case is a two-character parameter that is passed to an application to tell it which command has been given, and thus what process should be done.

Configure. To configure a system is to set it up, specifying certain parameter values or characteristics to complete the information that is needed so that it can run.

Context. A context is an application that has been started under CM.

Current context. The current context is the context you can see on the screen and interact with.

Default value. A default value is a value assigned automatically if you do not specify one.

Done. This status term indicates that the last command you invoked in the Executive has been completed.

Eliminating a context. To eliminate a context is to choose it in the list of "Contexts you can return to" and press ACTION-FINISH. This method discards work done in this context in this session.

Finishing a context. To finish a context is to carry out the usual steps to finish and save within that application.

Foreground. The current context, which owns the screen, runs in foreground.

Function keys. The function keys are keys f1 through f10 across the top of the keyboard. In CM, they can be assigned during a session or preassigned to applications.

Install. The term "install" has two meanings: to place software on a system, and to load a program from disk storage to workstation memory. In this manual, the second meaning has been used throughout.

Partition. A partition is a defined portion of workstation memory.

Preassigned function key. A preassigned function key is one assigned to an application through the CM Editor. Preassigned keys appear on the function key menu whenever CM is installed.

Running. This status term indicates that the context shown to its right is running.

Status terms. Status terms, which appear to the left of the items in the list of "Contexts you can return to," tell whether each context is running, waiting, done, swapped, or stopped.

Stopped. This status term indicates that the context shown to its right is in background, but is not running because it is allowed to run only in foreground.

Suspended. A context that has been swapped to disk (stored temporarily in a hard disk file) is said to be suspended because it is not actually running.

Swapped. This status term indicates that the context shown to its right has been stored temporarily on a hard disk. This context is active although suspended.

Swap to disk. To swap a context to disk is to place it temporarily in a designated swap file on a hard disk. A swapped context is still active (can be recalled with a keystroke) but is suspended (not actually running).

Switching contexts. To switch contexts is to press certain keys to cause one context to be replaced on the screen by another.

Type-ahead buffer. Characters that cannot yet be shown on the screen are stored in the type-ahead buffer until it becomes possible to display them.

User Configuration file. The User Configuration file contains general information needed to run the system for a given user.

Video pointer map. The video pointer map is an array of pointers, one for each line of the screen, that always point to the location of the associated line of the application's screen.

Waiting. This status term indicates that the context shown to its right is waiting for input.

INDEX

- Abbreviation field, in
 - CM Editor, 3-14
- ACTION key, 2-10, 3-2
- ACTION-FINISH, 2-6, 2-13
- ACTION-fn, 2-1, 2-5, 2-8, 2-9, 2-10
- ACTION-GO, 2-1, 2-5
- Application character map, 4-1 to 4-3
- Applications
 - assuming call from
 - Executive, 3-20
 - not swapped, 4-5
 - requiring parameters, 2-9, 3-19
 - vs. commands, 3-12
- Applications you can start, 2-1, 2-4, 2-8, 2-13, 3-12, 3-18
- Arrow keys, 2-1
- Assigning function keys in CM, 2-4, 2-7 to 2-9

- Background, 1-3, 2-2, 2-10, 2-11, 4-4
- Bullet, 2-5, 2-10
- Busy-wait loop, 4-3

- CANCEL
 - in CM, 2-1, 2-5, 2-10
 - in CM Editor, 3-12, 3-15
- Check (f4), 3-9, 3-12, 3-17
- CM Configuration file, 3-1, 3-3, 3-7, 3-8, 3-11, 3-21
- CM Configuration File Editor, **see** CM Editor
- CM Editor, 3-3, 3-7 to 3-18
 - starting, 3-8
- CmConfigEditor.run, 3-3
- CmConfig.sys, 3-3, 3-11
- CmInstall.run, 3-2
- CmNull.run, 3-2, 4-5
- Cm.run, 3-2
- Command case definition, 3-15
- Command Case field, in
 - CM Editor, 3-15, 3-20
- Command editing area, in
 - CM Editor, 3-12
- Command list area, in
 - CM Editor, 3-18
- Command Name field, in
 - CM Editor, 3-9, 3-12
- Commands
 - creating, 3-12
 - editing, 3-12, 3-13
 - name length, 3-13
 - removing, 3-12
 - renaming, 3-13
 - vs. applications, 3-12
- Communication between applications, 4-5
- Communication with CM, 4-5
- Communications applications, and swapping, 2-12, 3-4, 4-5
- Communications ports, sharing, 4-1
- Configuring CM, 3-1
- Context, definition, 1-2, 1-3, 2-4
- Contexts you can return to, 2-1, 2-5, 2-10, 2-12, 2-13
- Create (f5), 3-9, 3-11, 3-13
- Creating a command, example, 3-22
- Current context, 2-5, 2-10
 - definition, 1-3

- Deinstallation of CM, 3-2
- Done, 2-2, 2-11

Editing a command,
 example, 3-24
 Editing methods, 3-8
 Eliminating a context,
 2-6, 2-13
 Entry errors, in CM
 Editor, 3-25
 Executive, 2-9, 3-7,
 3-19, 4-5
 as a command, 3-20
 commands as
 applications, 3-12
 finishing, 2-6, 2-12
 status terms for, 2-2
 Exit run file, 4-5
 Exiting CM Editor, 3-18

Files required for CM,
 3-2 to 3-3
 Finish Executive, 2-6,
 2-12, 4-5
 FINISH in CM Editor,
 3-18
 Finishing a context,
 2-6, 2-11, 2-12
 Foreground, 1-3, 2-2,
 2-10, 2-11, 4-4
 Function Key field, CM
 Editor, 3-14
 Function key menu,
 in CM, 2-7
 in CM Editor, 3-16
 Function keys, 2-10
 assigning, 2-7 to 2-9
 default assignment,
 2-9
 in CM, 2-1
 in CM Editor, 3-9
 preassigned, 2-7 to
 2-9, 3-14

GetpStructure, 4-2 to
 4-3
 Graphics programs under
 CM, 4-1

Highlight
 in CM, 2-4, 2-5, 2-10
 in CM Editor, 3-8,
 3-14, 3-17

Input/error line, in CM
 Editor, 3-11, 3-18
 Installation
 of CM, 2-4, 3-2, 3-6
 to 3-7
 of system services,
 3-2
 Interprocess Communica-
 tion facility (IPC),
 4-5

Keyboard, ownership of,
 1-2

LockVideo, 4-2 to 4-3
 Logout, 2-6, 2-13, 3-2,
 3-12, 3-13
 from Executive, 2-13
 Low-memory interrupt
 vector table, 4-4

Memory, 1-2, 2-11, 3-3,
 3-7, 3-16
 requirements, 4-7
 Memory area entries,
 example, 3-23
 Memory area, in CM
 Editor, 3-15
 Memory (f1), 3-9, 3-15
 Memory Required field,
 CM Editor, 3-14
 Message area, in CM,
 2-1, 2-5, 2-8, 2-11
 Message line, in CM
 Editor, 3-11

NotifyCM, 4-6
 Number of Partitions
 field, in CM Editor,
 3-16

Parameters
 applications that
 require, 2-9
 passing, 3-20

- Partition overhead, 4-7
- Partition Sizes field,
 - in CM Editor, 3-16
- Partitions, memory, 1-2, 2-11, 3-3, 3-7, 3-16
- PosFrameCursor, 4-4
- Positioning cursor, 4-4
- Preassigned function
 - keys, 2-7 to 2-9, 3-14
- Programs
 - compatible with CM, 4-1
 - requiring alteration, 4-1
- Real-time applications
 - and swapping, 2-12, 3-4, 4-5
- Remove (f6), 3-9, 3-13, 3-17
- Removing a command,
 - example, 3-24
- Rename (f7), 3-9, 3-13, 3-17
- Renaming a command,
 - example, 3-25
- Run File Name field, in CM Editor, 3-13
- Running, 2-2, 2-10
- Saving, 2-6, 2-12, 2-13
- Screen areas, in CM Editor, 3-11 to 3-18
- Screen map, writing to, 4-2
- Screen
 - CM, 2-4, 2-5
 - CM Editor, 3-8, 3-9, 3-11
 - ownership of, 1-2, 1-3
- SetIntHandler, 4-4
- SetSwapDisable, 4-5
- SetTrapHandler, 4-4
- Show (f3), 3-9, 3-14
- Starting an application, 2-1, 2-4, 2-7
- Status terms, 2-2, 2-10 to 2-11
 - Done, 2-2, 2-11
 - Running, 2-2, 2-10
 - Stopped, 2-2, 2-11
 - Swapped, 2-2, 2-11
 - Waiting, 2-2, 2-10
- Suspended context, 1-3, 2-2, 2-12, 3-4
- Swap (f8), 3-9, 3-18
- Swap file, 3-1, 3-3, 3-4, 3-9, 3-18
 - creating, 3-5
 - size of, 3-5 to 3-6
- Swapped, 2-2, 2-11
- Swapping to disk, 1-2, 1-3, 2-5, 2-6, 2-11, 3-1, 3-3 to 3-4, 4-4
 - communications applications, 2-12, 3-4
 - real-time applications, 2-12, 3-4
- Switching contexts, 1-2, 1-3, 2-1, 2-5, 2-7, 2-8, 2-10
- System services,
 - writing, 4-8
- Target Memory field, in CM Editor, 3-16
- Undo (f2), 3-9, 3-17
- UnLockVideo, 4-2 to 4-3
- User Configuration file, 3-1, 3-2, 3-6, 3-7, 3-8
- Video Access Method (VAM), 4-2
- Video Display Management (VDM), 4-2
- Video output, 4-1 to 4-3
- Video pointer map, 4-1 to 4-3
- Waiting, 2-2, 2-10

USER'S COMMENT SHEET

Context Manager Manual

A-09-00382-01-A DT-600

We welcome your comments and suggestions. They help us improve our manuals. Please give specific page and paragraph references whenever possible.

Does this manual provide the information you need? Is it at the right level? What other types of manuals are needed?

Is this manual written clearly? What is unclear?

Is the format of this manual convenient in arrangement, in size?

Is this manual accurate? What is inaccurate?

Name _____ Date _____

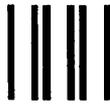
Title _____ Phone _____

Company Name/Department _____

Address _____

City _____ State _____ Zip Code _____

*Thank you. All comments become the property of
Convergent Technologies, Inc.*



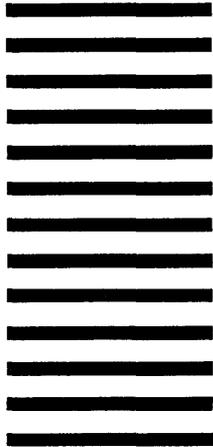
NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT #1309 SANTA CLARA, CA

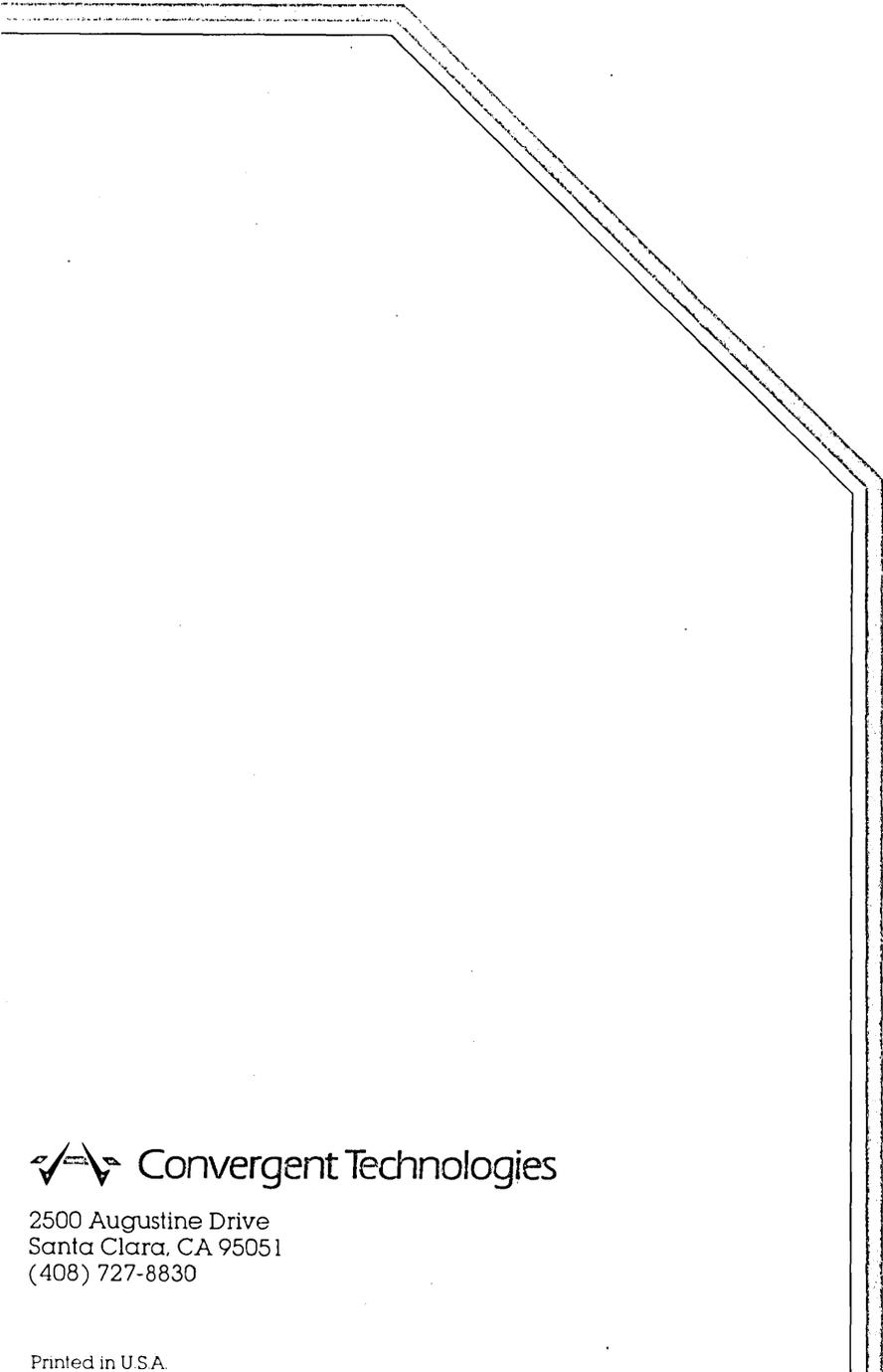
POSTAGE WILL BE PAID BY ADDRESSEE

Convergent Technologies
2500 Augustine Drive
Santa Clara, CA 95051

Attn: Technical Publications



Fold Here



 **Convergent Technologies**

2500 Augustine Drive
Santa Clara, CA 95051
(408) 727-8830