Burroughs

Reference Manual

B 20 Systems Context Manager

(Relative to Release Level 1.0)

Distribution Code SA

Priced Item
Printed in U.S.A.
March 1985

5016181

Burroughs

Reference Manual

B 20 Systems Context Manager

(Relative to Release Level 1.0)

Distribution Code SA

Priced Item
Printed in U.S.A.
March 1985

5016181

Burroughs cannot accept any financial or other responsibilities that may be the result of your use of this information or software material, including direct, indirect, special or consequential damages. There are no warranties extended or granted by this document or software material.

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

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

Comments or suggestions regarding this document should be submitted on a Field Communication Form (FCF) with the CLASS specified as 2 (S.SW: System Software), the Type specified as 3 (DOC), and the product specified as the 7-digit form number of the manual (for example, 5016181).

LIST OF EFFECTIVE PAGES

Page	Issue
Title	Original
ii	Original
iii	Original
iv	Blank
v thru x	Original
1-1 thru 1-3	Original
1-4	Blank
2-1 thru 2-30	Original
3-1 thru 3-14	Original
4-1 thru 4-14	Original
A-1 thru A-3	Original
A-4	Blank
B-1 thru B-3	Original
B-4	Blank
C-1 thru C-2	Original
D-1 thru D-2	Original
E-1 thru E-9	Original
E-10	Blank
1 thru 5	Original
6	Blank

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

LIST OF ILLUSTRATIONS

LIST OF TABLES

INTRODUCTION

Context Manager is a B 20 utility that allows several applications, utilities, and/or programs (referred to in this manual as applications) to run simultaneously on a B 20 multipartition operating system, version 5.0 or later. This manual contains descriptive and procedural information regarding the use of Context Manager.

To successfully use Context Manager, you should be familiar with:

- standard B 20 Executive commands
- material in the *B 20 Systems Operating System (BTOS) Reference Manual*
- manuals for the applications you plan to run under Context Manager

This manual consists of four sections and five appendixes:

Section 1	Provides you with an overview of the the utility, including its features and capabilities.
Section 2	Contains information on how to prepare and configure Context Manager. If you are setting up your own system, you should read this section as well as section 3.
Section 3	Discusses procedural information that enables you to operate Context Manager. If you are not configuring your system, reading this section provides you with all the information necessary to operate Context Manager.
Section 4	Contains information for the programmer writing applications to run under Context Manager. If you are writing such programs, you should read this section.

Appendix A Contains the error codes and messages you could receive while operating Context Manager.

Appendix B Describes the memory requirements necessary to run Context Manager.

Appendix C Tells you how to determine the number and sizes of memory partitions on your workstation.

Appendix D Describes how to rebuild your operating system to accommodate more contexts.

Appendix E Contains a glossary.

Additional, related information is available in the following manuals:

B 20 Systems Operating System (BTOS) Reference Manual
B 20 Systems Standard Software Operations Guide
B 20 Systems Editor Reference Manual
B 20 Systems Custom Installation and Reference Manual
B 20 Systems Programmer's Guide, Part 1

SECTION 1

CONTEXT MANAGER DESCRIPTION

Context Manager is a software utility that joins with BTOS to provide an environment in which several interactive applications, utilities, and/or programs can run simultaneously. This means that while you are working with an application on the screen, other applications, although not displayed, can be running in the background.

For example, if you must interrupt 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 word processing while the Copy command executes. You do not have to wait because each application remains active. Context Manager does not have to save, or open and close files during switching.

This section provides you with an overview of the Context Manager utility, its features, and how it works.

To perform operations with Context Manager, you should be familiar with the B 20 Operating System (BTOS) and the Executive commands. For further information, refer to the the following manuals:

- *B 20 Systems Operating System (BTOS) Reference Manual*
- *B 20 Systems Standard Software Operations Guide*

You should also be familiar with material regarding the various applications you plan to run under Context Manager.

NOTE

Throughout this manual, two terms are used with specialized meaning. The term application refers generally to applications, utilities, and/or programs you can access through Context Manager. The term context refers to an active (or open) application.

FEATURES

Context Manager includes the following features:

- it starts applications from Context Manager's own display screen
- it uses a Context Manager Configuration File Editor (CM Editor) for editing its configuration file
- it has no fixed set of applications
- it can run more than one application concurrently, including two or more of the same application
- it can run your own applications

Context Manager has a single distinct screen to which you can return at any time. On the right side of the screen is a list of applications that you can start. On the left side of the screen is a list of applications you already opened (called contexts) with a corresponding status term.

Applications can be BTOS-based or they can run under operating systems that BTOS hosts (for example, MS-DOS or CP/M-86).

Context Manager does not have a fixed set of applications. You can configure your system to run whichever applications are important to you, and because of Context Manager's flexibility, you can add your own application to it with little or no special programming.

HOW CONTEXT MANAGER WORKS

Several partitions make up workstation memory under Context Manager. BTOS resides in the low end of memory; Context Manager itself remains in the primary partition. As you begin each application, Context Manager places it in one of the remaining partitions. (The person configuring Context Manager determines the total number of partitions and their respective sizes; refer to section 2 for details on system configuration.)

The most recently started context has control of the screen and keyboard. When you switch from context A to context B, for example, control of the screen and keyboard passes to context B. Other contexts can also run, and they write whatever they would normally display on the screen to an area in memory.

The number of partitions available does not limit the number of applications you can start. Given sufficient disk space, you can swap an application to a hard disk (suspend it temporarily in a disk swap file). Context Manager can handle as many as ten contexts at once. In this manner, Context Manager can swap contexts back and forth from disk as you need them.

SECTION 2

INSTALLING AND CONFIGURING CONTEXT MANAGER

Before you can use Context Manager, you must perform several operations to prepare Context Manager to run the specific applications, utilities and/or programs that you want to use on your B 20 system.

This section describes how to load Context Manager from the distribution diskette into your system's memory and how to set up the program to fit your needs. It includes information and procedures for:

- installing the Context Manager software
- determining how to invoke Context Manager
- creating a swap file to increase the number and ease of operation of the applications available through Context Manager
- configuring Context Manager using the CM Editor

To perform the operations described in this section, you must be familiar with the B 20 Operating System (BTOS) and the Executive commands. For information, see the *B 20 Systems Operating System (BTOS) Reference Manual* and the *B 20 Systems Standard Software Operations Guide.*

You also need specific information about each application that you add to Context Manager (for example, run file name, memory size, and command case value). For this information, see the documentation for the particular application.

INSTALLING THE CONTEXT MANAGER SOFTWARE

The Context Manager software operates only on B 20 Family of systems that have a minimum memory of 512K, a hard disk (at the master workstation), and a multipartition operating system.

To install the Context Manager software, use the following procedure:

1. Sign on to your system.

2. Set the Path to [sys]<sys>. (For information about setting the Path, see the *B 20 Systems Standard Software Operations Guide.*)

3. Insert the Context Manager distribution diskette in drive [f0].

   CAUTION

   This distribution diskette is write-protected. Do not write-enable it, nor use it as a working copy.

4. In the Executive Command field type **Software Installation.**

5. Press **GO.**

6. Respond to the prompts that appear for installing the software on your particular system. You must indicate if you have a B 21, B 22, B 25 or an XE520 system, and if your workstation is a standalone, master, or cluster. A screen message notifies you when the installation of the Context Manager software is complete.

7. Remove the Context Manager diskette and store it in a safe place.

FILE REQUIREMENTS OF CONTEXT MANAGER

To operate successfully, Context Manager requires the files listed in table 2-1. After you install the Context Manager software, these files exist in the system directory on your hard disk.

Table 2-1. Context Manager Files

File Name	Use
CmInstall.run	This is the run file that installs Context Manager.
Cm.run	This is the Context Manager run file.
CmNull.run	This is the null exit run file which is used for communication with Context Manager when an application finishes.
CmConfig.sys	This is a sample Context Manager Configuration file. You can edit this file or create other configuration files to conform to the applications you want to invoke from Context Manager. You must create a separate configuration file with a unique name for each cluster workstation. CmConfig.sys is the default file name. You can alter this name and make it specific for a particular user. (See Configuring Context Manager with the CM Editor, in this section.)
CmEditor.run	This is the CM Editor run file. This special editor allows you to edit a Context Manager Configuration file(s).
CM.user	This is a sample user file. If you type the user name **CM** at the SignOn form, this file allows Context Manager to load automatically. You can edit this file or create other user files for automatic signon purposes. (See Access Through the SignOn Form, in this section.)

Table 2-1. Context Manager Files (continued)

File Name	Use
	NOTE If the file [sys]<sys>Cm.user already exists on your hard disk, it is not overwritten during the software installation procedure.
CmDeInstall.run	This file deinstalls Context Manager at the time of a debugger crash.

DETERMINING HOW TO INVOKE CONTEXT MANAGER

Context Manager is not an installed system service. You can log out of the program and continue to run your system without rebooting.

NOTE

You must install all system services (for example, Install Spooler, Install Queue Manager, etc.) before invoking Context Manager.

You have the option of invoking Context Manager in one of two ways:

- through the Executive
- through the SignOn form, bypassing the Executive

You must determine how you want to access the program and, if necessary, set up your system accordingly. The following paragraphs describe each of these access methods.

Access Through the Executive

You can access Context Manager through the Executive by entering **Install Context Manager** in the Executive Command field and pressing **RETURN.** The Install Context Manager form appears. In the [Configuration file name] field, you enter the name of the Context Manager configuration file you want to use. If you want to use the default configuration file, you type **Install Context Manager** in the Executive Command field and press **GO.**

Access Through the SignOn Form

You can set up your system so that Context Manager loads automatically (bypassing the Executive) after you sign on at the B 20 SignOn form. You do this by creating a User file or editing the Cm.user file which exists on your hard disk after you install the software.

The file Cm.user is a sample file that allows you to sign on to Context Manager directly by typing **CM** in the User Name field of the SignOn form.

If you have an XE520 system, you must insure that the correct number of dollar directories are created for each workstation. To do this, use the following procedure the first time you sign on to Context Manager after software installation:

1. Type **user** in the User Name field of the SignOn form. The Command field of the Executive appears.

2. Type **logout** at the Command field; the SignOn form reappears.

3. Type **CM** in the User Name field of the SignOn form.

You can edit the Cm.user file to change the user name or create other user files by using the User File Editor or the Editor. For more information about these editing operations, see the *B 20 Systems Custom Installation and Reference Manual* and the *B 20 Systems Editor Reference Manual.*

For direct access to Context Manager through the SignOn form, each user file must conform to the following sample:

UserName.user

```
:SignOnVolume:Sys
:SignOnDirectory:sys
:SignOnFilePrefix:
:SignOnPassword:
:SignOnExitFile:[sys]<sys>SignOn.run
:SignOnChainFile:[sys]<sys>CmInstall.run
'Install Context Manager'
[sys]<sys>filename
:ExecCmdFile:[sys]<sys>sys.cmds
```

The filename (in bold on the sample) default is CmConfig.sys. You should replace this with the name of the configuration file you create with the CM Editor that specifies the applications you want to use through Context Manager. (See Configuring Context Manager with the CM Editor, in this section.)

NOTE

For XE520 systems, the correct :ExecCmdFile: directory is <user> or <cmd>.

SWAP FILES

When you open an application file through Context Manager, it places the open application (called a context) in a memory partition. The number of contexts available to you depends on the number and sizes of the memory partitions. You can increase the number of available contexts by creating a hard disk file called a swap file. If you open an application when all partitions are in use, Context Manager temporarily places a context in this swap file. The transfer of the context to a hard disk swap file is called swapping. Context Manager can swap to a swap file on a local hard disk or over a cluster to a master workstation hard disk.

Although Context Manager works whether you create a swap file on the hard disk or not, swapping increases the flexibility of Context Manager by allowing you to open up

more applications than the number of partitions specified at the CM Editor screen (see Memory Area in this section). If you do not create a swap file, you are limited to one context for each memory partition (normally two to three).

Swapping contexts is faster than starting and stopping a context. Both the contexts in partitions and the contexts in a swap file are immediately available to you without waiting for save, open, or close file operations.

Swapped contexts are suspended, not active. When you request a swapped context, Context Manager clears a partition by swapping another context to the swap file and then swaps the requested context into the cleared partition.

Context Manager chooses a context to swap based on the partition size needed and the status of the contexts. It does not swap communications (such as the Asynchronous Terminal Emulator) or real-time applications (such as a context in direct printing mode).

Creating a Swap File

To create a swap file on a hard disk, use the following procedure:

1. In the Executive Command field, type **Create File.**

2. Press **RETURN.** The following form appears:

```
Create File
  File name                                    _____
  [Volume or Directory password]               _____
  [File password]                              _____
  [File protection level (default = 15)]       _____
  [Size in sectors (default = 0)]              _____
  [Overwrite ok?]                              _____
```

NOTE

You can use the **RETURN, NEXT, Up Arrow,** and/or **Down Arrow** keys to move the highlight through the form. (For more information about this form, see the *B 20 Systems Standard Software Operations Guide.*)

3. Type the complete file name (including volume and directory) in the File Name field.

   To swap over a cluster, precede the volume name with an exclamation point. For example, the entry [sys]<sys>ABCswap creates the file ABCswap on the system directory on the local disk named sys. The entry [!sys]<sys>ABCswap creates the file ABCswap on the system directory on the clustered hard disk named sys.

   Use a unique swap file name for each cluster station. The name can include the user name for the workstation, for example, [sys]<sys>EricsSwapFile or [d0]<sys>EricsSwapFile if the cluster workstation has local file storage.

4. If the volume or directory is password-protected, move the highlight to the [Volume or Directory password] field and type the password. The password entry appears as a series of # characters.

5. If the volume or directory is password-protected and you want this file to be password-protected, move the highlight to the [File password] field and type the password.

6. If you want to enter the file protection level, move the highlight to the [File protection level (default = 15)] field and type the new level.

7. Determine the number of sectors required for the swap file (based on an estimate of how many contexts you may swap to the hard disk during a session).

   Each 1K byte requires two sectors. Therefore, a 200K-byte application requires 400 sectors, and a swap file large enough for five 200K-byte applications must have 2000 sectors.

   The minimum number of sectors required is the sum of the sectors needed for each application run through Context Manager at a workstation.

   The maximum number of sectors needed is 10 times the size of the largest application (10 is the maximum

number of contexts Context Manager can actively manage). You should use this calculation as a guide only.

See appendixes B and C for more information.

8. Type the number of sectors for the swap file in the [Size in sector (default = 0)] field.

9. Press **GO**. The system creates the swap file. When you edit a Context Manager Configuration file, you identify the swap file associated with it by pressing the **Swap** (**f8**) key and entering the complete swap filename. (See Assigning a Swap File, in this section.)

CONFIGURING CONTEXT MANAGER WITH THE CM EDITOR

Context Manager contains a default Context Manager Configuration file which includes three applications (Logout, Executive, and WRITEone). Before you use Context Manager, you edit this file and/or create other configuration files to provide the Context Manager with the specific details it needs to function. You edit or create configuration files with the CM Editor, a separate program that allows you to supply information about the workstation memory, number and sizes of memory partitions, and applications you want Context Manager to run.

You can specify different configuration files for different users. If you create several Context Manager Configuration files, each must have a distinct name. Any User file you create should contain the name of that user's Context Manager Configuration file. (See Access Through the SignOn Form, in this section.)

Entering the CM Editor

To enter the CM Editor through the Executive, use the following procedure:

1. Type **CM Editor** (or an abbreviation that makes this command name unique) in the Executive Command field.

2. To open a Context Manager Configuration file as you enter the CM Editor, press **RETURN.** The following form appears:

   CM Editor
   [Configuration file name] _____

3. Type the configuration file name. If this file does not already exist, a message appears asking if you want to create a file with this name. The default for this field is [sys]<sys>Config.sys.

4. You then have two options:

   - If you want to call up this file or create a file with this name, press **GO.** The CM Editor screen appears with the highlight on the Command Name field and the configuration file name in the half-bright area of the Identification Line. (See figure 2-1.)

   - If you want to discard your configuration file name entry, press **CANCEL.** The highlight remains on the [Configuration file name] field, so you can enter another name.

The CM Editor Screen

The CM Editor screen contains the following areas:

- Identification Line
- Message Line
- Input/Error Line
- Command Editing Area
- Memory Area
- Function Key Display
- Command List Area

Details about these areas appear in the subsections that follow. Figure 2-1 shows a sample CM Editor screen with each area labeled.

On the CM Editor screen, you can edit any highlighted field in any order; you do not have to complete all fields in one editing session. You use the **Memory** function key (**f1**) to move the highlight from the Command Name field of the Command Editing Area to the Memory Area. You use **RETURN**, **NEXT**, **Up Arrow**, or **Down Arrow** to move the highlight from field to field within an area.

Within a highlighted field, the **Left Arrow**, **Right Arrow**, **Spacebar**, or **BACK SPACE** move the cursor horizontally, and

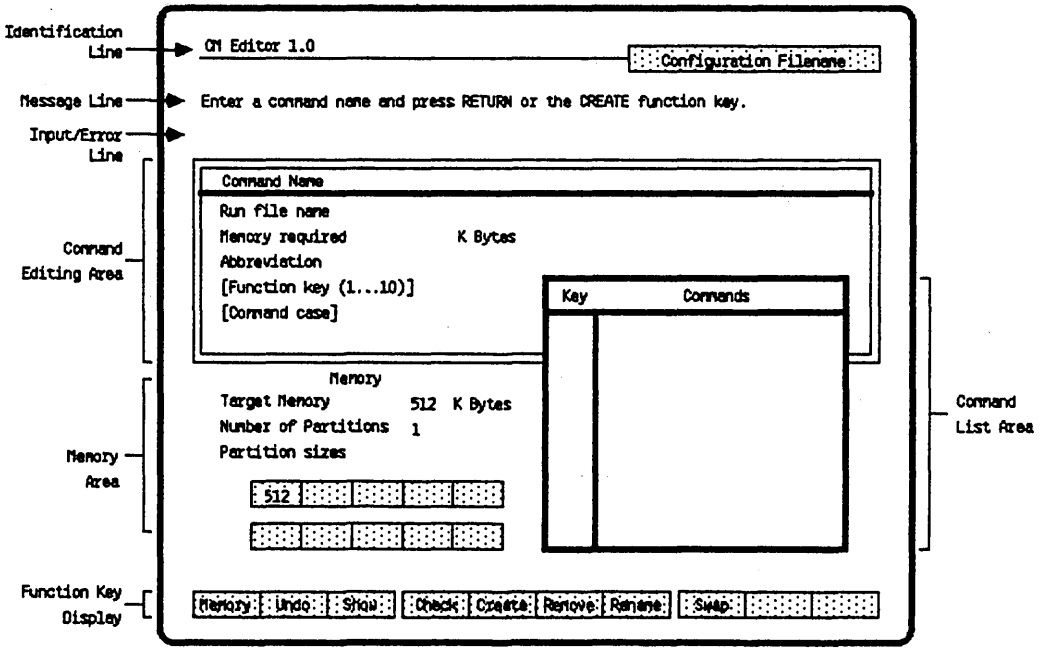

Figure 2-1. CM Editor Screen

the usual editing modes (insert or overtype) and functions (**DELETE, COPY,** etc.) operate. By holding down **CODE** and then pressing **DELETE,** you can remove the entire entry from a field. For more information about Executive key operations, see the *B 20 Systems Standard Software Operations Guide.*

Identification Line

The top line on the CM Editor screen is the Identification Line. On the left side of this line is the program name and version number. On the right is a half-bright field which displays the name of the configuration file you are editing.

Message Line

The line immediately below the Identification Line on the CM Editor screen is the Message Line. During your operation of CM Editor, messages display on this line telling you what you can or should do next.

When the CM Editor screen first appears, the following message is on this line if you entered a Context Manager Configuration file name in the CM Editor form:

Enter a command name and press RETURN or the Create function key.

The highlight appears in the Command Name field of the Command Editing Area (see Command Editing Area, in this section).

If you accessed the CM Editor without providing a file name, you see the following on the Message Line:

Please enter a configuration file name.

The highlight appears on the Input/Error Line with the default file name, [sys]<sys>CmConfig.sys (see Input/Error Line, in this section).

Input/Error Line

The Input/Error Line appears below the Message Line. It serves as an input field when the Message Line prompts you for entries. The Input/Error Line also displays the results of the **Check** function key operation (see Function Key Display, in this section) and error messages that occur during an editing session. These error messages are self-explanatory.

When the CM Editor screen first appears, if the Message Line prompts you to enter a Context Manager Configuration file name, the Input/Error Line is highlighted and contains the default file name, [sys]<sys>CmConfig.sys. You can accept this default name by pressing **GO**, or you can edit the file name and then press **GO**.

If you enter a Context Manager Configuration file name that does not exist and press **GO**, a message asks for confirmation that you want to create a file with this name. You press **GO** to confirm, or **CANCEL** to discard this name.

NOTE

If you want to create a new Context Manager Configuration file, you must enter an unused configuration filename. CM Editor does not check nor prompt you that a file exists. It overwrites an existing file with an identical name.

Command Editing Area

The Command Editing Area of the CM Editor screen is within a rectangular box surrounded by a double line. It contains fields which you use to add applications (such as Multiplan or B 20 Mail Manager) and Executive commands (such as Floppy Copy or Files) to the CM Editor Command List. Table 2-2 describes each of these fields. Context Manager can recognize 18 commands that you define through the CM Editor, although it can manage only a maximum of 10 contexts simultaneously.

Table 2-2. Command Editing Area Fields

Field	Description
Command Name	In this field, you specify the name of the application you wish to add to, edit, or remove from the Command List.
Run file name	In this field, you specify the name of the run file for the application named in the Command Name field.
Memory required	In this field, you enter the amount of memory in K bytes that this application requires.
Abbreviation	In this field, you specify an abbreviation for the name of this application. If you assign a function key to this application, this abbreviation appears on the Function Key Display in the Context Manager screen. The default value is the first six characters of the Command Name field entry.
[Function Key (1...10)]	If you wish to preassign a function key to this application, you enter the number of an available function key in this field. The Command List Area displays the function key numbers which are already assigned. You do not have to make an entry in this field.
[Command case]	If you can invoke this application under more than one condition (for example, you can invoke WRITEone through Recover or SignOn), you must enter a command case value in this field. This value is a two-character parameter that tells the application which command was given. For the correct case value, see the documentation for the application. The default is 00.

Table 2-2. Command Editing Area Fields (continued)

Field	Description
	If you must call this application from the Executive, you enter **CM** in this field. See Applications You Can Start from Context Manager, in this section.

You also use the Command Editing Area to edit, remove, or rename existing commands. The CM Editor refers to both Executive commands and applications as commands; however, the Context Manager screen calls them both applications.

If you press **CANCEL** when the highlight is on any of the fields in this area, the system discards all your entries or edits in the Command Editing fields.

Memory Area

The Memory Area is below the Command Editing Area on the left side of the CM Editor screen. It contains fields in which you enter memory and partition information. Table 2-3 describes each of the fields in the Memory Area.

You enter this area by pressing the **Memory** (**f1**) function key (see Function Key Display, in this section). If you press **CANCEL** when the highlight is on any of the fields in this area, the system cancels your entries and the highlight returns to the field it was in when you pressed the **Memory** key.

Table 2-3. Memory Area Fields

Field	Description
Target Memory	In this field, you specify in K bytes the memory size of your workstation. You can edit the default value (512K) displayed in the field.

Table 2-3. Memory Area Fields (continued)

Field	Description
Number of Partitions	In this field, you specify the number of partitions into which you want the workstation memory divided. The default value is 1. The target memory size and the sizes of the applications you want to run determine this number. The maximum number of partitions allowed is 10, but the usual number is one to three. See appendix C for more information about determining partitions.
Partition Sizes	This field consists of two rows of five blocks each. Each block represents one possible partition. You specify the size of each partition in K bytes according to the sizes of the applications you want to run. The minimum size is 15 K bytes. A default (230) appears in the first block. See appendix C for more information about determining partition sizes.

Function Key Display

At the bottom of the CM Editor screen is the Function Key Display, a set of temporary labels that correspond to the function keys on your keyboard. Only the functions that appear on the key labels are available during your operation of the CM Editor. Table 2-4 describes the effect of each function key.

Table 2-4. CM Editor Function Keys

Key	Function	Description
f1	**Memory**	When you press **Memory**, the system moves the highlight to the Target Memory field of the Memory Area.
f2	**Undo**	When you press **Undo**, the system replaces the value in the highlighted field with the immediately previous value for that field. In numeric fields, if the previous value was blank, the value becomes 0.
f3	**Show**	When you press **Show**, the system overwrites the CM Editor Function Key Display with the current values on the Context Manager screen Function Key Display (see section 3).
f4	**Check**	When you press **Check**, the system validates your entries. It reports (on the Input/Error Line) any discrepancies.
f5	**Create**	When you press **Create**, the system records in the Command List Area the name displayed in the Command Name field.
f6	**Remove**	When you press **Remove**, the system removes the name in the Command Name field from the Command List Area.
f7	**Rename**	When you press **Rename**, the system allows you to edit the command name displayed in the Command Name field.
f8	**Swap**	When you press **Swap**, the system prompts you to identify the swap file for this configuration file.

Command List Area

The Command List Area is within a box on the right side of the CM Editor screen that partially overlays the Command Editing Area. This area displays the commands currently assigned to this particular configuration file. (On the Context Manager screen, these commands are called Applications You Can Start.) You do not edit the Command List. It records commands entered in the Command Editing Area during this or previous editing sessions.

NOTE

If you enter the CM Editor from an Executive running under Context Manager to edit your own configuration file, the changes you make do not display on the Applications You Can Start list of the Context Manager screen until you log out and reinstall Context Manager.

Applications You Can Start from Context Manager

The design of each application determines how it runs from Context Manager. Context Manager allows you to start applications in two ways:

- directly from Context Manager
- indirectly through the Executive

Applications Accessed Directly from Context Manager

An application that you can access directly from Context Manager must meet two requirements:

- You must be able to access the application without supplying parameters.
- The application must start by clearing the screen.

Examples of applications that meet these requirements are B 20 Mail Manager, and Multiplan. When you define

these commands, you specify their run files in the Run File Name field. (See Adding a New Command to a Context Manager Configuration File, in this section.)

Applications Accessed Indirectly Through the Executive

You can access applications that do not meet the screen clearing or parameter requirements indirectly through the Executive. The Executive meets both of these requirements.

Examples of applications that you access through the Executive are Floppy Copy, Files, and Copy.

When you specify the commands for these applications in the Command Editing Area of the CM Editor screen, you type **[sys]<sys>Exec.run** in the Run file name field and the case value, **CM**, in the [Command case] field.

When you request these applications at the Context Manager screen, Context Manager calls the Executive run file. The command case value, CM, tells the Executive that Context Manager called the application, and the Executive receives control of the specified command. The Executive displays the command form; you complete the form and press **GO**. The Executive then returns control to Context Manager. (For more information, see section 3.)

NOTE

To define the Executive as a command, you leave the [Command case] field blank.

Adding a New Command to a Context Manager Configuration File

Context Manager can recognize 18 commands. To add a new application to the Command List, use the following procedure:

1. In the Command Name field, type the name of the application exactly as you want it to appear in the list of Applications You Can Start on the Context Manager screen (for example, Multiplan).

You can enter up to 50 characters, however only 32 characters display on the CM Editor Command List and the Context Manager screen.

Some applications produce unpredictable results when used through Context Manager, and therefore, are called dirty. If this application is dirty, type an asterisk at the end of the command name (for example, **Forms Editor***). This signals Context Manager to suspend the program in the background. See section 4 for a discussion of clean and dirty applications.

2. Press **RETURN** or **Create** (**f5**). The highlight moves to the Run file name field.

NOTE

You can press **GO** at this point instead of **Create**; the new command name displays in the Command List area, without an entry in the Run file name field. This operation allows you to make a partial entry. The CM Editor assumes that you will re-enter this field and supply the run file name before you try to use the command in Context Manager. You cannot operate the command unless the correct run file is present. The Check operation (see Function Key Display, in this section) does not verify the existence of the run file.

3. You have two options at the Run file name field:

   - If Context Manager can call this application directly, type the run file name of the application.
   - If Context Manager must call the application through the Executive, type **[sys]<sys>Exec.run.**

For more information about applications you can call directly or indirectly, see Applications You Can Start from Context Manager, in this section.

4. Press **RETURN.** The highlight moves to the Memory required field.

5. Type the memory required for the application in K bytes. (See appendix B for memory requirements for B 20 applications, and section 4 for estimation instructions of memory requirements for other applications.)

NOTE

You should be sure that the entry you make in this field is accurate. Context Manager uses this figure to place the context in the proper partition.

6. Press **RETURN.** The highlight moves to the Abbreviation field. The default entry (the first six characters of the Command Name) appears in this field. You can edit this entry, if you wish.

   When you assign a function key to the command, this abbreviation appears in the Context Manager Function Key Display.

7. Press **RETURN.** The highlight moves to the [Function Key (1...10)] field. If you wish to preassign a function key to this command, check the Key column at the left of the Command List Area, and then enter the number of an available function key. This assignment is optional; if a key is not assigned to this application at this time, Context Manager automatically assigns it the next available function key when you start the application at the Context Manager screen. (See section 3 for more information.)

   To display the preassigned function keys over the CM Editor Function Key Display, press **Show (f3)**. Press **CANCEL** to return the CM Editor Function Key Display.

NOTE

Do not assign all 10 function keys. Leave several blank so you can start applications that do not have preassigned function keys.

8. Press **RETURN**. The highlight moves to the [Command case] field. The default command case value is 00. If you can invoke this application through more than one command, you must enter a command case value which indicates to the application which command was given (for example, WRITEone requires a CM command case). For the correct parameter value, see the documentation for the application.

   For more information about command case values, see New Command in the *B 20 Systems Standard Software Operations Guide.*

9. Press **GO** to create this command. The command name and the function key number (if assigned) appear in the Command List Area.

10. If you are creating a new configuration file, you must provide memory and partition information in the Memory Area of the CM Editor screen. To enter the Memory Area, press the **Memory (f1)** function key. See Editing the Memory Area, in this section for instructions.

You can now add other commands to the Command List, by repeating the above procedures.

CAUTION

You must include the command **Logout**, with the run file [sys]<sys>SignOn.run, on the Command List of each Context Manager Configuration file you create.

Editing the Memory Area

In the Memory Area of the CM Editor screen, you provide memory and partition information for your system. You can enter this area at any time during an editing session by pressing **Memory (f1)**. If you press **CANCEL** at any time while you are in the Memory Area, the highlight returns to the field where you were before you entered this area.

To edit memory allocations, use the following procedure:

1. Press **Memory** (f1). The highlight moves to the Target Memory field. If you are editing this area for the first time, a default value appears in the field.

2. To delete the existing value, hold down **CODE** and press **DELETE** or press the **BACK SPACE** key repeatedly.

3. Type the memory size (in K bytes) of the workstation for which you are configuring this file.

4. Press **RETURN.** The highlight moves to the Number of Partitions field. The default value for this field is 1.

5. Enter the number of memory partitions for this workstation. The total memory size of the workstation and the sizes of the applications that you want to run govern this number. The maximum number is 10, but usually, you should create one to three partitions on a system. If you create more partitions than the system can support, the following message appears:

   This version of the OS cannot support any more contexts.

   See appendix C for more information about determining partition sizes and appendix D for a discussion of system build.

6. Press **RETURN.** The highlight moves to the first block in the field under Partition Sizes. This field has two rows divided into five blocks each; one block represents each possible partition. The first block contains a default value.

7. Type the size in K bytes of each partition. You can use up to four digits. Each partition size should reflect the size of an application you want to run in that partition. The minimum size is 15K. See appendix C for information on determining these sizes.

   Use the **Left** and **Right Arrow** keys to move the edit cursor within a block. The CM Editor allows you to

access only the number blocks that you specified in the Number of Partitions field.

8. When your Memory Area entries are complete, press **GO** to record the changes and exit this area. The highlight moves to the field it was on when you entered the Memory Area.

   If you press **CANCEL** instead of **GO**, the highlight moves to the field it was on when you entered the Memory Area, and the changes are not recorded.

Assigning a Swap File

If you created a swap file on your hard disk, you must assign it to a configuration file (see Swap Files, in this section).

Use the following procedure to assign a swap file:

1. Enter the CM Editor by typing **CM Editor** in the Executive Command field.

2. Press **RETURN.** The CM Editor form appears.

3. In the [Configuration file name] field, type the name of the configuration file that you want the swap file associated with, or use the default file.

4. Press **GO.** The CM Editor screen appears with the highlight on the Command Name field.

5. Press the **Swap** (**f8**) function key. The highlight moves to the Input/Error Line and the Message Line instructs you to enter a swap file name.

6. Type the name of the swap file in the Input/Error field.

7. Press **GO.** The system records the swap file as associated with this configuration file.

   If you press **CANCEL** instead of **GO**, no swap file assignment takes place.

Changing a Command

After you create a Context Manager Configuration file, you can use the CM Editor to change the commands you configured for the file. The CM Editor provides three command editing options:

- You can edit existing command information by entering the command name and changing Command Editing Area fields.
- You can use Remove function (**f6**) to remove a command from the Command List.
- You can use Rename function (**f7**) to change the name of a command on the Command List.

The following paragraphs describe each of these options in detail.

Editing a Command

The CM Editor allows you to change all or part of the command information of a Context Manager Configuration file. You make these changes in the Command Editing Area of the CM Editor screen. While making changes to any field, you should adhere to the parameter restrictions for that field. See Command Editing Area and Adding a New Command to a Context Manager Configuration File, in this section for information on field parameter restrictions.

CAUTION

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

To edit command information, use the following procedure:

1. Enter the CM Editor by typing **CM Editor** in the Executive Command field.
2. Press **RETURN.** The CM Editor form appears.

3. In the [Configuration file name] field, type the name of the configuration file you want to edit.

4. Press **GO.** The CM Editor screen appears with the highlight on the Command Name field.

5. In the Command Name field, type the name of the command you want to change. If you enter a command name that is not identical to one on the Command List, the CM Editor creates your entry as a new command name.

6. Press **GO.** The information previously entered for this command appears in the fields of the Command Editing Area, and the highlight moves to the first field of the area.

7. You can then edit any of the command fields, using the **RETURN, NEXT, Up Arrow, Down Arrow** or the appropriate function keys to move the highlight from field to field. You hold down **CODE** and press **DELETE** to remove all of an existing entry at once, or use the **DELETE** and **BACK SPACE** keys to remove all or part of an existing entry.

8. To record your changes, press **GO** while the highlight is in any field. If you changed the Command Name or a function key number, the change appears in the Command List Area.

Removing a Command

To remove an existing command from a Context Manager Configuration file, use the following procedure.

CAUTION

Do not remove the **Logout** command from the Command List of a Context Manager Configuration file.

1. Enter the CM Editor by typing **CM Editor** in the Executive Command field.

2. Press **RETURN.** The CM Editor form appears.

3. In the [Configuration file name] field, type the name of the configuration file from which you want to remove a command.

4. Press **GO.** The CM Editor screen appears with the highlight on the Command Name field.

5. Type the name of the command you want to remove in the Command Name field.

6. Press the **Remove** function key (**f6**). (The highlight can be on any field on the Command Editing Area when you press this key.) A message appears on the Message Line asking you to confirm this deletion.

7. Press **GO.** The command name is removed from the list.

   If you press **CANCEL** instead of **GO**, the Remove operation does not take place, and the command remains on the Command List.

Renaming a Command

To rename an existing command, use the following procedure:

1. Enter the CM Editor by typing **CM Editor** in the Executive Command field.

2. Press **RETURN.** The CM Editor form appears.

3. In the [Configuration file name] field, type the name of the configuration file in which you want to rename a command.

4. Press **GO.** The CM Editor screen appears with the highlight on the Command Name field.

5. Type the existing name of the command you want to rename in the Command Name field.

6. Press the **Rename** (**f7**) function key.

7. Delete the existing command name using the **CODE-DELETE** key combination or the **BACK SPACE** key.

8. Type the new name for this command in the Command Name field.

9. Press **GO**. The highlight moves to the Run file name field. You can edit this or any of the other command fields, if necessary (see Editing a Command, in this section).

10. Press **GO** to record the command name change. The new name replaces the old one on the Command List.

    If you press **CANCEL** instead of **GO**, the Rename operation does not take place, and the old command name remains on the Command List.

Exiting the CM Editor

When you complete an editing session, you exit the CM Editor by using the following procedure:

1. Press **FINISH**. The following occurs:

   - A message appears on the Message Line asking you to confirm that you want to finish this session.
   - The Input/Error Line informs you if any inconsistencies exist in your entries.

2. At this step you have two options:

   - If the Input/Error Line indicates that inconsistencies exist, press **CANCEL**. The CM Editor screen remains, and you can then edit your entries to eliminate these inconsistencies.
   - If no inconsistencies exist, press **GO** to exit the CM Editor and overwrite the configuration file with the edits of this session.

NOTE

Your editing changes do not become part of the configuration file until you exit the CM Editor by pressing **FINISH** and then **GO** in accordance with the above procedure. However, if you accessed the CM Editor from within Context Manager, your changes do not take effect until you logout and then invoke Context Manager again using the edited configuration file.

DISPLAYING THE CONTENTS OF A CONTEXT MANAGER CONFIGURATION FILE

If you want to look at the contents of a Context Manager Configuration file, you can use the **Type** command in the Executive to display the contents of the file on your screen. Table 2-5 shows an example of a Context Manager Configuration file containing two commands (the Executive and Logout). A series of entries comparable to the bold text in this table should exist for each command you define with the CM Editor.

Table 2-5. A Sample Context Manager Configuration File

```
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
```

Table 2-5. A Sample Context Manager Configuration File (continued)

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

:CommandName:Logout
:RunFileName:[sys]<sys>SignOn.run
:CommandAbbreviation:Logout
:MemorySize:100
:CommandCase:00
:FunctionKey:
```

SECTION 3

OPERATING CONTEXT MANAGER

Once you install and configure Context Manager to run applications, you use the procedures described in this section to run these applications simultaneously through Context Manager. Instructions contained in this section include procedures necessary to:

- start the system
- open applications
- swap contexts
- close contexts
- exit Context Manager

Throughout this section, two terms are used with specialized meaning: applications and contexts. The term application refers generally to applications, utilities, and/or programs you can access through Context Manager. The term context refers to an active (or open) application.

STARTING THE SYSTEM

You can access Context Manager directly through the initial SignOn form or through an Executive command. The method you use depends upon how you configure your system (refer to Access Through the SignOn Form, in section 2 of this manual). One of the following procedures invokes Context Manager:

- Accessing Context Manager from the SignOn form

  1. Type **CM**.
  2. Press **GO**.

  The Context Manager screen appears.

- Invoking Context Manager through the Executive

  1. At the Executive command prompt, type **Install Context Manager**

  2. Press **GO.**

  The Context Manager screen appears.

THE CONTEXT MANAGER SCREEN

Figure 3-1 illustrates a sample Context Manager screen with each area labeled. Context Manager's screen includes the following areas:

- Applications You Can Start
- Contexts You Can Return To
- Status terms
- Message area
- Function Key Display

This screen lists the programs you can run using Context Manager; it also indicates their current status and gives you prompts regarding the operation you are performing.

To move the cursor throughout the Context Manager screen, you use the Arrow keys. The **Up** and **Down Arrow** keys move the cursor vertically through the screen; the **Right** and **Left Arrow** keys move the cursor horizontally across the screen.

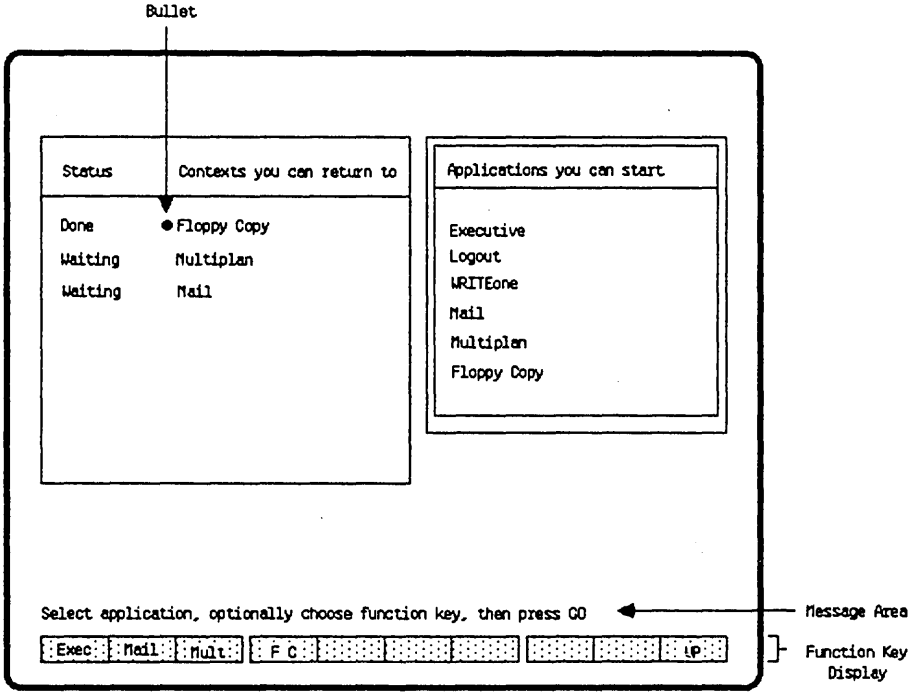

Figure 3-1. Context Manager Screen (Sample)

Applications You Can Start

In this manual, an application is a general term describing applications, utilities, and/or programs you can run using Context Manager. A list of applications available to you appears in the block on the right of the screen under the heading Applications You Can Start. (Refer to section 2 for information regarding configuration.)

The Context Manager release disk includes the ability to invoke the Executive, WRITEone, and Logout applications. These commands come with the default configuration file. Additionally, Context Manager retains color and graphics qualities between contexts.

The Executive application refers to the Executive command level, and it allows you to enter Executive commands through Context Manager. The WRITEone application gives you word processing capabilities, and the Logout application allows you to exit Context Manager.

NOTE

The WRITEone software is not on the release disk; the release disk contains only the command to invoke it.

Contexts You Can Return To

Contexts are open (active) applications. The Contexts You Can Return To column lists the applications you opened in the order that you opened them.

The system indicates the current context by placing a bullet to the left of it.

Status Terms

The Status column to the left of the bullet indicates the status of the corresponding context. Status terms can be any one of the following:

- Running
- Waiting
- Done
- Swapped
- Stopped

Table 3-1 describes these terms.

Table 3-1. Status Terms

Term	Description
Running	The application is currently operating.
Waiting	The application is not operating; it is waiting for your input. This message does not appear for applications you enter through the Executive application (such as Floppy Copy or Files).
Done	Your last operation is complete. This message appears for Executive applications only (such as Floppy Copy or Files).
Swapped	The system temporarily stores this context on disk to make room for another context. The swapped context is open and available to you, but is suspended until you recall it.
Stopped	The context is in the background, but not running. This context can only run in foreground.

Message Area

The Message area is just above the Function Key Display at the bottom of your screen. It prompts you with the status, options, and/or limitations associated with the context you are working on.

Function Key Display

The highlighted strip at the bottom of the Context Manager screen is the Function Key Display; a set of temporary labels on these keys corresponds to the function keys on your keyboard. On your system, these labels may all be blank, or some may show abbreviations for specific applications which are listed under Applications You Can Start. These labels can be configured differently for each user (refer to section 2 for details regarding configuration).

If labels already appear on your Function Key Display, they were preassigned during configuration for frequently used applications. This preassignment saves time: Context Manager does not have to name these function keys every time you begin a work session.

Preassigned function keys remain intact from work session to work session; function keys that Context Manager assigns last only for the current work session.

SELECTING APPLICATIONS AND CONTEXTS

You select applications to open, and contexts to work on by using the Arrow keys and the function keys labeled **f1** through **f10** on your keyboard.

Arrow Keys

Arrow keys move the highlight horizontally from the Applications You Can Start area to the Contexts You Can Return To area; arrow keys also move the highlight vertically through these fields.

The **Left Arrow** and **Right Arrow** keys move the highlight on the screen between the block labeled Applications You Can Start and the block labeled Contexts You Can Return To.

The **Up Arrow** and **Down Arrow** keys move the highlight vertically through each of these blocks, or fields.

Function Keys

Across the top of your keyboard are ten function keys labeled **f1** through **f10**.

While you use Context Manager, the function keys on your keyboard correspond to the highlighted Function Key Display at the bottom of the screen. When you begin your work session, labels on your Function Key Display can be blank, or some can show abbreviations for applications available to you. You use function keys together with the **ACTION** key to select applications or contexts on which to work.

OPENING AN APPLICATION

Before you can open an application and begin a work session, the application must have a function key assigned to it. Function keys are assigned in two ways:

- You preassign applications to function keys during configuration (refer to section 2).
- Context Manager assigns each application to a function key when you open the application.

Opening an Application with a Preassigned Function Key

When the application you choose has a preassigned function key, you open the application by pressing the preassigned function key and then pressing **GO**.

The Context Manager screen automatically highlights the application you chose and its corresponding function key on the Function Key Display.

The Message area responds with **Loading...**, and the screen of the context appears. Now you can begin your work session.

To return to the Context Manager screen at any time without closing the context, you press **ACTION** and **GO (ACTION-GO)**.

The Context Manager screen overlays the context you are working on; its name then appears under Contexts You Can Return To, along with a Status term.

To return to a context, you press **ACTION** and the designated function key (**ACTION-Fn**), and continue your work session.

You can have one or more of the same type of application open at any time, two Executives for example. Each has its own function key, and you work with each as distinct contexts.

Opening an Application Using a Function Key Assigned by Context Manager

Context Manager assigns an application to a function key if a key was not preassigned. You use the following procedure to designate a function key through Context Manager.

1. Use the **Up** or **Down Arrow** keys to highlight the application you want to open.

2. Press **GO**.

The Context Manager screen automatically highlights the application you chose and assigns the application to the next available function key on the display, starting from the left on the Function Key Display.

The Message area then reads **Loading...**, and the context's screen appears. Now you can begin your work session.

To return to the Context Manager screen at any time without closing the context, you press **ACTION-GO**.

The Context Manager screen overlays the context you are working on; its name then appears under Contexts You Can Return To, along with a Status term.

You can have one or more of the same type of application open at any time, two Executives for example. Each can be assigned to a different function key by Context Manager, and you work with each as two distinct contexts.

STARTING AN APPLICATION THAT REQUIRES PARAMETERS

To open an application that requires parameters, such as Copy, use the following procedure:

1. Use the **Up** or **Down Arrow** keys to highlight the application you want to open.

2. Press **GO**.

3. Fill in the required parameters in the form that appears on the screen.

4. Press **GO**.

The command executes and you can begin your work session.

STARTING A SECOND APPLICATION

To open a second application, follow one of the procedures noted in this section for opening an application (refer to Opening an Application with a Preassigned Function Key and/or Opening an Application Using a Function Key Assigned by Context Manager).

You return to the Context Manager Screen by pressing **ACTION-GO**; then you again use one of the methods noted above to open a second application.

Depending on the kind of system you have, the message area could tell you at this point (or after you open several other applications) that a context is being swapped to disk. This means that the context is being stored temporarily on the hard disk to make room for another context. The swapped context is still open, but suspended until you recall it (refer to Swapping a Context, in this section).

The message area then reads **Loading...**, and the screen of the second context appears. Now you can begin your work session.

To return to the Context Manager screen without closing the context, you press **ACTION-GO**. The context you worked on appears under Contexts You Can Return To, along with any other contexts and Status messages.

CONTEXTS YOU CAN RETURN TO

You can return to a context from another context or from the Context Manager screen.

To return to a context from another context, you press **ACTION-Fn**).

To return to a context from Context Manager screen, you can use either of the following procedures:

1. Use the Arrow Keys to move the highlight to the context you want, under the list of Contexts You Can Return To.

2. Press **GO**.

The screen of the chosen context displays.

OR

1. Press the designated function key for the context you want to access.

2. Press **GO**.

The screen of the chosen context displays.

NOTE

When you switch from context to context, the first context moves from the foreground to the background of memory until you recall it.

SWAPPING A CONTEXT

A swapped context refers to a context that you temporarily put in your system's swap file; this leaves room for you to open more applications when your system is full. The swapping capability and the extent of swapping is dependent upon the size of the configured swap file (refer to section 2, in this manual).

If your system does not have this feature, you receive the message: **In order to start another application, you must have a swap file.** This means that you must close a context before you can start another application.

If you configure your system to have this feature, you receive one of the following messages when you exceed the number of contexts your system can hold:

- **Swapping contexts...**
- **Warning: you may not be able to swap a context.**
- **The swap file is full.**

Table 3-2 describes these messages.

Once swapped, the context is still open; you can return to it again by pressing **ACTION-Fn.**

NOTE

Context Manager does not swap two types of contexts: communications applications such as Asynchronous Terminal Emulator, or real-time applications such as a context in direct printing.

Table 3-2. Swapping Messages

Message	Description
Swapping contexts...	This message tells you that the system is storing a context. When you receive the message, **Done**, the swap is complete. You can continue to input while awaiting this message; the system stores characters until space is available. Your new context screen displays this input.

Table 3-2. Swapping Messages (continued)

Message	Description
Warning: you may not be able to swap a context.	This warning tells you that your swap file may not hold another context. You should close a context before opening another application.
The swap file is full.	This message alerts you that the swap file is full. You must close a context to continue using the swap file.

CLOSING A CONTEXT

When you close a context you can either save or discard your work session.

Saving the Work Session

To close the context and save the work session, you close the context in the usual way, thereby saving the work session.

After closing the context, the Context Manager screen appears. The message area reads, **Finishing...** and the context disappears from the list under Contexts You Can Return To, and reappears under Applications You Can Start.

Discarding the Work Session

To close a context and discard your work session, close the context from the Context Manager screen using the following procedure:

1. Press **ACTION-GO** to invoke the Context Manager screen.

2. Using the **Up** and **Down Arrow** keys, highlight the context you want to close.

2. Press **ACTION-FINISH.**

The Message area reads **Finishing....** The finished context disappears from the list under Contexts You Can Return To and reappears under Applications You Can Start.

EXITING CONTEXT MANAGER

You can exit Context Manager from either the Executive or Logout applications, as noted in this section. Before you can exit Context Manager, however, you must close all contexts. You can use one of the following procedures to exit Context Manager.

Exiting from the Executive

To exit Context Manager from the Executive application, you use the following procedure:

1. Close all contexts (refer to Closing a Context, in this section).

2. Open the Executive application.

3. When the Executive command prompt appears, type **logout.**

4. Press **GO.**

The Context Manager screen appears, and the Message area reads, **Logging out....** The SignOn form then appears on your screen.

Exiting Using Logout

To exit Context Manager through the Logout application, you use the following procedure:

1. Close all contexts (refer to Closing a Context, in this section).

2. Using the **Up** and **Down Arrow** keys, highlight the Logout application.

3. Press **GO**.

The Message area reads, **Logging out...** and the SignOn form appears on the screen.

SECTION 4

PROGRAMMER'S NOTES

Context Manager operates on a B 20 system multipartition operating system, version 5.0 or later. From the application programmer's viewpoint, Context Manager acts as an extension of the operating system and takes over some of its functions. It allows applications to run concurrently, each using a separate memory partition.

This section describes how applications interact under Context Manager and also provides you with programming information necessary to customize these applications to your needs. It includes the following information:

- application modifications under Context Manager
- suspended applications
- behavior of BTOS products and system services under Context Manager
- swappability of applications
- communication with Context Manager and between applications
- estimating memory requirements
- handling video output

The information described in this section presumes your familiarity with the B 20 operating system (BTOS). For information regarding the system, refer to the following manuals:

- *B 20 Systems Operating System (BTOS) Reference Manual*
- *B 20 Systems Programmer's Guide, Part 1*

You should also be familiar with the documentation for each application that you add to Context Manager.

APPLICATIONS YOU MUST MODIFY TO RUN UNDER CONTEXT MANAGER

Context Manager acts as an extension of the operating system. Thus, with the installation of Context Manager, most existing applications run without modification.

You must modify the following applications so they can run under Context Manager:

- any application that writes directly to the screen map in memory
- any application containing busy wait loops that do not allow the processor to give up the keyboard when the application is waiting to read keystrokes
- an application that changes a value in the low-memory interrupt vector table directly in memory
- an application that positions the cursor using the video controller port
- an application that changes the exit run file

In addition, you can run only one graphics application at a time under Context Manager; graphics output does not go to the character map--it goes to the graphics board.

Similarly, two applications cannot share communications ports. For example, you cannot run two versions of the Asynchronous Terminal Emulator at the same time--both want to use channel A.

Applications That Write Directly to the Screen Map

The mapping of video lines insures that the output from 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, applications that use VAM and VDM calls work automatically.

Context Manager cannot detect applications that write directly to the screen; the output from their write requests goes directly to the screen, whether or not they own the screen lines.

The 5.0 and later versions of the operating system provide three System Common procedures that you can use to allow your applications to work:

- LockVideo
- UnLockVideo
- GetpStructure

For further information regarding the use of these procedures, refer to the *B 20 Systems Operating System (BTOS) Reference 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 the parameter structcode=7. The returned pointer aims at the video pointer map.

When you want to move characters to the screen later in the application, you lock all video structures. Locking keeps other applications from changing video structures while you are using them. You invoke locking by a call to LockVideo. To do the actual write, you must have a structure that looks like a single line of the video, and base it on the appropriate pointer entry in the video pointer map. When you finish your write, you must unlock the structures by calling UnLockVideo. You perform the series of locking, writing, and unlocking on a line-by-line basis.

These procedures insure that two applications do not write to the screen simultaneously.

NOTE

If you use these calls, Context Manager allows the application to run in background. Without these calls, it suspends the application and gives the status **Stopped.**

Busy Wait Loops

You should not write application code using a busy wait loop when dealing with the keyboard.

The usual way to read keystrokes from the keyboard is to issue a BTOS ReadKbd. An application using calls to ReadKbdDirect with mode 1 can generate a busy wait. Since mode 1 means that the operating system returns immediately whether a character is available or not, you can write an application to loop, awaiting a character to arrive.

When an application is running in foreground, Context Manager raises that application's priority (i.e., gives it a lower numeric value). When an application runs in background, however, its priority returns to normal. If an application contains a busy loop and is running in foreground, its priority is higher than those of other applications in background. Because the foreground application always runs, background applications do not get processor time; consequently, background applications stop running.

Low-Memory Interrupt Vector Table

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

- SetIntHandler
- SetTrapHandler

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

To allow your application to be swapped out, you use SetTrapHandler. For example, you can swap your application if the handler you are providing is a software interrupt

service routine; no interrupts of this type can occur while the application is swapped out. (For further information, refer to the *B 20 Systems Operating System (BTOS) Reference Manual.*)

Positioning the Cursor

Since an application 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, you use the BTOS procedure PosFrameCursor to position the cursor.

Exit Run File

When you choose an application to start from Context Manager, Context Manager chains to the run file and sets the exit run file to CmNull.run. Therefore, CmNull.run runs when the application is complete. CmNull.run sends a message to Context Manager asking it to terminate the context. (Refer to Communication with Context Manager and Between Applications, in this section.) Upon receiving this message, Context Manager terminates the context that sent it.

If you write an application that changes the exit run file, Context Manager never gets the message that the application ended. In this case, you must provide a way to run CmNull.run.

For example, if the Executive sets itself as the exit run file, the Executive reloads only after its application finishes. The new command, **EXIT EXECUTIVE**, allows you to get back to Context Manager from the Executive. This command invokes CmNull.run.

APPLICATIONS SUSPENDED IN BACKGROUND

A context is in one of the four states noted in table 4-1. This denotation is necessary to avoid unpredictable results

for applications that write directly to the screen or move the cursor using the video controller port, etc. Context Manager suspends the context if it is not on the screen and marks it as either tentatively or absolutely dirty.

Table 4-1. Context States

State	Description
Tentatively clean	Contexts start in this state.
Tentatively dirty	Contexts in the tentatively clean state go to this state if they do a ResetVideo.
Absolutely clean	Contexts go into this state if they do any of the following: GetPStructure InitVidFrame NotifyCM (msgType=8)
Absolutely dirty	Contexts go into this state if they do NotifyCM (msgType=9)

NOTE

The procedure calls in table 4-1 do not contain complete parameter lists.

To mark an application explicitly as dirty (not allowing it to run in background), you use the CM Editor and add an asterisk (*) to the end of the command(s) you want to mark (see section 2).

For example, if you want to mark the Executive as dirty, you can press the **Rename** function key (**f7**) on the CM Editor screen and change the name to Executive*. Thereafter, Context Manager treats the application as though it had sent a message saying it was absolutely dirty.

NOTE

The detection of dirty applications is not foolproof; unpredictable results (such as multiple cursors and overlapping text from several applications) can occur unless you change applications that are undetectable to Context Manager.

BEHAVIOR OF BTOS PRODUCTS UNDER CONTEXT MANAGER

Table 4-2 describes the status of Burroughs' applications with regard to Context Manager. The following terms and their definitions define an application's behavior.

- Clean indicates no side effects of the application running under Context Manager.
- Dirty means that the application either accesses video directly, addresses the cursor through a port, changes color or font, and causes unpredictable results when run in background.
- Busy Wait refers to applications that poll the keyboard or other devices. It causes all other applications to suspend.

Table 4-2. Product Behavior Under Context Manager

Product	Status
2780/3780 RJE 4.0	Clean
B 20 Mail Manager 1.0	Clean
BISYNC/3270 Enhanced 4.0	Dirty
BGP 4.0	Clean
Graphics 4.0	Clean*

* One version per workstation

Table 4-2. Product Behavior Under Context Manager (continued)

Product	Status
CM Editor 1.0	Clean
CP/M-86 1.0.1	Dirty
Editor 5.0	Clean
Executive 5.0	Clean
Forms Editor 4.0	Dirty
Font Designer 4.0	Dirty
ISAM 4.0	Clean
MS-DOS 2.11 (BIOS 2.0)	Dirty
Multiplan 3.0	Clean#
SNA/3270 4.0	Dirty
SNA/RJE 1.0	Clean
Sort/Merge 4.0	Clean
User File Editor 4.0	Clean
WRITEone 3.0.6	Clean

Access directly through CM or through the Executive (not through a submit routine).

Table 4-2. Product Behavior Under Context Manager (continued)

Compilers/Interpreters	
BASIC Interpreter 4.0	Clean
BASIC Compiler 4.0	Clean
COBOL 4.0	Clean
FORTRAN 4.0	Clean
Pascal 4.0	Clean
X.25 4.0	Clean

NOTE

Graphics Library 4.0 or later is clean; you should relink graphics applications with this library if they are to work with Context Manager. You can run only one graphics application at a time under Context Manager.

CP/M-86 and MS-DOS are application-relative. A dirty application run under them makes them dirty.

SWAPPABILITY OF APPLICATIONS

Only those applications that do not have outstanding requests can swap. For applications that have outstanding requests to swap, you must change system services. To respond to BTOS swap requests, they must either finish servicing the application request or return the request to

the OS for requeuing when the application swaps back in. Depending on the timing and the ability of the system service to respond to a swap request for a client, an application can be swappable at one time and not swappable at another. For information on Handling swap requests, refer to the *B 20 Systems Operating System (BTOS) Reference Manual.*

Applications That Cannot Be Swapped

Certain calls to BTOS, such as SetCommIsr and SetIntHandler, identify an application as one that cannot swap. Real-time and communications applications that include such calls are recognized as unswappable. Therefore, Context Manager never tries to swap them to disk.

If you want to insure that your application is not swapped, you can add a call to the BTOS routine SetSwapDisable. (For further information on this routine, refer to the *B 20 Systems Operating System (BTOS) Reference Manual.*)

Before you invoke Context Manager, you must install any system service you intend to use. The following table indicates which BTOS system services respond to swap requests.

Table 4-3. Behavior of BTOS System Services Under Context Manager

Product	Handles Request
B 20 Mail Manager 1.0	Yes
ISAM Server 4.0	No
Queue Manager 5.0	Yes
Spooler 5.0	Yes

NOTE

For information regarding swapping requests on systems that run Context Manager, refer to the *B 20 Systems Operating System (BTOS) Reference Manual.*

COMMUNICATION WITH CONTEXT MANAGER AND BETWEEN APPLICATIONS

Applications can communicate with Context Manager using the Interprocess Communication facility (IPC) in BTOS. (For more information on the use of IPC, refer to the *B 20 Systems Operating System (BTOS) Reference Manual.*

Context Manager sits at an exchange waiting for the user to type an **ACTION**-keystroke or for a message from an application. A message from the Executive to Context Manager signaling your last command is an example of this procedure.

The operating system call, NotifyCM, affects this communication. Its definition is:

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

The following table describes appropriate msgType values.

Table 4-4. Work Value Codes, Originators, and Messages for NotifyCM

Work Value Code	Messages	Originator
1	Terminate context; pass erc	CmNull
2	New command	Executive
3	Logout	SignOn
4	Terminate context	Executive
5	Install Context Manager	CmInstall

Table 4-4. Work Value Codes, Originators, and Messages for NotifyCM (continued)

Work Value Code	Messages	Originator
6	Graphics application	Graphics Application
7	Terminate context; pass error message	CmNull
8	Context runs in background (clean)	Any Application
9	Context cannot run in background (dirty)	Any Application
10	QueryGraphics. Context Manager returns the context handle (user number) of the context which is currently running a graphics application, or 0FFFFh if none is running.	Executive

The parameters pbMsg and cbMsg are only valid for msgType 1, 2, 4, and 7 messages.

Work codes 8 and 9 need further explanation. Depending on the sequence of calls that an application makes to video routines, Context Manager marks the context as clean or dirty (able or unable to be run in background). Any application 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 Context Manager can communicate with each other using their own definition of messages through the IPC facility.

ESTIMATING MEMORY REQUIREMENTS

When you configure the system, you must supply Context Manager with the memory requirement for each defined

application. To obtain an application's memory requirement, you use the following procedure:

1. Obtain the run file size from the map file that Linker generates. The last entry in the second column of the map file is hexadecimal and is the size in bytes. In the following example, the correct entry is in boldface.

Start	Stop	Length	Name	Class
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	

2. Convert this hexadecimal value to a decimal.

3. Add the partition overhead (10240).

4. Examine all calls to AllocMemoryLL and AllocMemorySL, then find the total memory that the application could allocate while running.

5. Add this total to the value in step 2.

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

The number you arrive at in step 5 is the memory requirement for that particular application.

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

If an application conforms to the amount of memory available, (that is, calls AllocAllMemorySL), you should create as large a partition as is practical to accommodate this run file. Then you identify a run file size to fit the partition.

HANDLING VIDEO OUTPUT

The video pointer map is a system structure for each memory partition. It is made up of an array of pointers, one for each line of the screen, that always points to the location of the associated line of the application's screen. Memory partitions also have individual character maps. The application character map is an array that stores the lines of the screen when an application is running in background.

When you switch from context A to context B, the following occurs:

- Application A's application character map copies the real screen.
- The real screen receives a copy of application B's character map.
- In both video pointer maps, each pointer updates to point to the appropriate position, either on the real screen or on an application character map. It is the action of these individual pointers that makes the Context Manager screen appear to overlay the current context's screen when you press **ACTION-GO**.
- Screen attributes, such as color, remain intact.

APPENDIX A

ERROR MESSAGES

The following error codes and messages aid you when you run interactive applications through Context Manager.

Error Code	Error Message
6	The master workstation went down.
203	A specified file does not exist.
204	A specified directory does not exist.
205	A file specification for this application is not valid.
215	The volume or device specified is not on line.
219	You must add the correct password to a file specification.
220	A file needed for this application is already in use.
400	Not enough memory in this partition to run this application.
801	This version of the OS cannot support any more contexts.
813	A context in memory cannot be swapped out.
2440	This application must be invoked through the Exec; edit the Config file.

Error Code	Error Message
40001	CM inconsistency, suggest you save all contexts.
40002	There are no more function keys available.
40003	There is nothing to finish.
40004	There are no active contexts.
40005	Internal error: Wrong exchange .
40006	Internal error: Region status inconsistency.
40007	Internal error: swap file inconsistency.
40008	The swap file is full -- cannot swap any more contexts.
40009	Internal error: Too many swaps.
40010	The run file is too large to run in any region.
40011	You cannot logout, there are active contexts.
40012	Swapping is not yet implemented.
40013	The Context Manager is already installed.
40015	In order to start another application you must have a swap file.
40016	You can only run one graphics application at a time.
40017	This hardware is unknown to the Context Manager.
40018	There is no current context.

Error Code	Error Message
40019	Internal error: Invalid line indices specified for a map switch.
40020	An existing context cannot be swapped out to start a new application.
40021	You must finish the highlighted context to return to the one you selected.
40022	To deinstall CM you must finish all contexts, then log out.
40023	The run file needed for this application does not exist.
40025	Internal error: Swap count invalid.
40026	Unable to access the swap file -- cannot swap contexts.

APPENDIX B

MEMORY REQUIREMENTS: CONTEXT MANAGER AND APPLICATIONS

Context Manager requires 65K bytes of memory space to operate successfully.

Context Manager requires a memory utilization figure for each application in the Context Manager Configuration file. Table B-1 lists the memory requirements in K bytes for B 20 applications. (These memory requirements are a guide; actual application requirements may vary.) A value of **all** in a column means that the application uses all available memory.

The applications marked with an asterisk require parameter input, and therefore, you must access them through the Executive. The memory requirements given for these applications are misleading. Since the Executive is loaded first, the required memory for applications run through it is the minimum amount required for the Executive. For example, if the memory requirement for an application you must access through the Executive is given as 50K, the application actually needs to run in a partition of at least 101K, the recommended minimum for the Executive.

Table B-1. Memory Requirements of B 20 Applications

Product (Minimum Version)	Min	Better Performance	Max
ATE	60	-	60
BISYNC/3270 4.0	86	-	86
Business Graphics (BGP) 4.0*	170	220	300
CM Editor 1.0	60	-	60

*Must be accessed through the Executive

Table B-1. Memory Requirements of B 20 Applications (continued)

Product (Minimum Version)	Min	Better Performance	Max
CP/M-86 1.0.1*	110	140	all
Editor 5.0	90	150	200
Executive 5.0	100	150	all
Forms Editor 4.0	56	-	56
Forms Reporter 4.0*	41	-	41
Font Designer 4.0*	70	-	70
ISAM 4.0			
Configure*	45	-	45
Copy*	85	120	all
Create*	41	-	41
Delete*	41	-	41
Rename*	41	-	41
Reorganize*	125	150	all
Set Protection*	41	-	41
Status*	95	-	95
Mail Manager 1.0	140	200	all
MS-DOS 2.11 (BIOS 2.0)*	140	256	all
MTE	96	-	178
Multiplan 3.0	80	120	160
SNA/3270 4.0	80	-	80
SNA/RJE 1.0	84	-	84
Sort/Merge 5.0			
Merge*	40	80	all
Sort*	60	100	all

*Must be accessed through the Executive

Table B-1. Memory Requirements of B 20 Applications (continued)

Product (Minimum Version)	Min	Better Performance	Max
User File Editor 5.0	80	-	80
WRITEone 3.0.6	230	300	all
Compilers/Interpreters			
BASIC Interpreter 4.0*	151	-	all
BASIC Compiler 4.0*	207	-	all
COBOL 4.0*	165	200	all
FORTRAN 4.0*	140	200	all
Pascal 4.0*	140	200	all

*Must be accessed through the Executive

APPENDIX C

DETERMINING THE NUMBER AND SIZES OF MEMORY PARTITIONS

When you edit the Context Manager Configuration file with the CM Editor, you must specify how you want your workstation memory divided into partitions. You calculate the number of memory partitions and the size of each partition.

Use the following procedure to determine partition numbers and sizes (table C-1 illustrates this procedure):

1. Determine the available memory for partitions on your workstation by subtracting the operating system size, the Context Manager size, and the sizes of any installed system services from the workstation memory size.

2. Decide how many partitions, n, you want.

3. Multiply this number, n, times 10K.

4. Subtract the result of step 3 from the partition memory (calculated in step 1) to allow for partition overhead. The result is the amount of target memory available for applications.

5. Divide the amount of target memory available for applications (calculated in step 4) into partition sizes.

6. Enter the results of your calculations in the Number of Partitions field and the Partition Sizes field in the Memory Editing Area of the CM Editor screen.

Table C-1. Sample Calculation of Memory Partitions for a Workstation

Environment: B 25 multipartition master system with 1MB running BTOS 5.0; Spooler and Queue Manager installed; user requires two partitions.

```
 1000K  (workstation memory size)
- 191K  (BTOS multipartition 5.0 size)
-  65K  (Context Manager size)
-  34K  (Spooler size)
-  18K  (Queue Manager size)
= 692K  (available partition memory)

-  20K  (two partition overhead)
= 672K  (actual usable partition memory)
```

This allows two partitions: one with 500K of memory, and one with 172K.

NOTE

You can divide the usable partition memory into the desired number and sizes of partitions.

APPENDIX D

REBUILDING A SYSTEM TO ALLOW MORE CONTEXTS

The number of partitions specified at system build using the nPartitions parameter in the sysgen prefix file determines the number of contexts that Context Manager can support simultaneously. The nPartitions parameter usually refers to the physical division of memory; however, Context Manager uses nPartitions as virtual entities. You specify the physical division of memory when you edit a Context Manager Configuration file. In the following discussion, nPartitions refers to virtual partitions.

As you create a context, a virtual partition is assigned to it. If the number of contexts the system supports is exceeded during the operation of Context Manager, the following message displays:

This version of the OS cannot support any more contexts.

You can rebuild your system to include more virtual partitions, and therefore more contexts, by modifying the prefix file of your operating system. To determine the prefix file for your workstation, see Build a Custom Operating System in the *B 20 Systems Programmer's Guide, Part 1.*

The prefix file contains a line similar to the following: %Set(nPartitions, 6). Only prefix files for multipartition operating systems (the operating system Context Manager requires) have this entry.

To determine the number of partitions you need, you add the number of partitions the operating system requires, the number of partitions Context Manager requires, the number of contexts you want, and the number of installed system services. Table D-1 illustrates this calculation.

You change the n in nPartitions to this sum. You must then assemble the sysgen files and link the BTOS version, following the directions in the *B 20 Systems Programmer's Guide, Part 1.*

Table D-1. Sample Calculation of nPartitions Value

Environment: User requires five contexts; three system services are installed. The number of virtual partitions to specify is:

```
 1   (operating system partition)
+1   (Context Manager partition)
+5   (contexts partitions)
+3   (installed system services partitions)
=10  (number of partitions needed)
```

APPENDIX E

GLOSSARY

Absolutely Clean Context State

Contexts go into the absolutely clean state if they do any of the following: GetPStructure (lprgpVidMem line), InitVidFrame, NotifyCM (notlMarkDirty).

Absolutely Dirty Context State

Contexts go into the absolutely dirty state if they do NotifyCM (lMarkDirty). If the context is not on the screen and is marked absolutely dirty, the context is suspended.

Application

In this manual, an application is a general term describing applications, utilities, and/or programs that you can run using Context Manager. A list of applications available appears in the block on the right of the Context Manager screen under the heading, Applications You Can Start.

Arrow Keys

Arrow keys move the highlight horizontally from the Applications You Can Start area to the Contexts You Can Return To area on the Context Manager screen, and they move the cursor horizontally within the fields of the CM Editor screen. These keys also move the highlight vertically within the areas of both screens.

Background

Any context other than the current context is said to be in background.

Bullet

On the Context Manager screen, the bullet is the small dot to the left of the name of the current context under Contexts You Can Return To.

Busy Wait Status

Busy Wait status refers to applications that poll the keyboard or other devices. It causes all other applications to suspend.

Check (f4) Key

When you press this key at the CM Editor screen, the system validates your entries. It reports (on the Input/Error Line) any discrepancies.

Clean Status

Clean status indicates no side effects of the program running under Context Manager.

CmConfig.sys File

CmConfig.sys is a sample Context Manager Configuration file. You can edit this file or create other configuration files to conform to the applications you want to invoke from Context Manager. CmConfig.sys is the default file name.

CmDeInstall.run File

CmDeInstall.run is the file that deinstalls Context Manager at the time of a debugger crash.

CM Editor

The CM Editor is a separate program that allows you to set up Context Manager to meet your needs. You use the CM Editor to edit the Context Manager Configuration file and/or create additional user-specific configuration files.

CmEditor.run File

CmEditor.run is the CM Editor run file. This special editor allows you to edit a Context Manager Configuration file(s).

CmInstall.run File

CmInstall.run is the run file that installs Context Manager.

CmNull.run File

CmNull.run is the null exit run file which is used for communication with Context Manager when an application finishes.

Cm.run.File

Cm.run is the Context Manager run file.

Cm.user File

Cm.user is a sample user file. If you type the user name CM at the SignOn form, this file allows Context Manager to load automatically. You can edit this file or create other user files for automatic signon purposes.

Command Editing Area

The Command Editing Area of the CM Editor screen contains fields which you use to add applications (such as Multiplan, or B 20 Mail Manager) and Executive commands (such as Floppy Copy or Files) to the CM Editor Command List. It appears within a rectangular box surrounded by a double line. You can also use this area to edit, remove, or rename existing commands.

Command List Area

The Command List Area is located within a box on the right side of the CM Editor screen that partially overlays the Command Editing Area. The Command List Area displays the commands currently assigned to a particular configuration file. On the Context Manager screen, these commands are called Applications You Can Start.

Context

A context is an active (open) application. The most recently started context has control of the screen and keyboard.

Context Manager

Context Manager is a software utility that allows several applications, utilities, and/or programs to run simultaneously on a B 20 multipartition operating system, version 5.0 or later.

Context Manager Files

Context Manager files are files required to operate Context Manager successfully. (See individual Glossary entries.)

Context Manager Release Disk (See Release Disk.)

Context States

There are four context states: tentatively clean, tentatively dirty, absolutely clean, absolutely dirty. (See individual Glossary entries.)

Create (f5) Key

When you press this key at the CM Editor screen, the system records in the Command List Area the name displayed in the Command Name field.

Default Value

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

Dirty Status

Dirty status means that the program either accesses video directly, addresses the cursor through a port, or changes color or font, and causes unpredictable results when run in background.

Done Status Term

A status of Done means your last operation is complete.

Down Arrow and Up Arrow Keys

The **Down** and **Up Arrow** keys move the highlight vertically through each of the areas on the Context Manager and CM Editor screens.

Executive Application

The Executive application refers to the Executive command level. It allows you to enter Executive commands through Context Manager. Applications that do not meet the screen

clearing or parameter requirements of Context Manager must be accessed indirectly through the Executive rather than directly through Context Manager.

Function Key Display

The Function Key Display is the highlighted strip at the bottom of the Context Manager and the CM Editor screen. It is a set of temporary labels corresponding to the function keys on your keyboard.

Function Keys

The ten Function keys are located across the top of your keyboard and are labeled **f1** through **f10.** While you use Context Manager or the CM Editor, the Function keys on your keyboard correspond to the highlighted Function Key Display at the bottom of the screen. Only the functions or applications (on the Context Manager screen) that appear on the key labels are available at any point during the operation of Context Manager or the CM Editor.

Identification Line

The Identification Line is the top line on the CM Editor screen. The program name and version number appear on the left side of this line. A half-bright field which displays the name of the configuration file you are editing appears on the right.

Input/Error Line

The Input/Error Line appears below the Message Line on the CM Editor screen. It serves as an input field when the Message Line prompts you for entries. The Input/Error line also displays the results of the Check function key operation and error messages that occur during an editing session.

Left Arrow and Right Arrow Keys

The **Left** and **Right Arrow** keys move the highlight on the Context Manager screen between the blocks labeled Applications You Can Start and Contexts You Can Return To. On the CM Editor screen, these keys move the cursor horizontally within a field.

Memory Area

The Memory Area is below the Command Editing Area on the CM Editor screen. It contains fields in which you enter memory and partition information.

Memory (f1) Key

When you press this key at the CM Editor screen, the system moves the highlight to the first field of the Memory Area, which is Target Memory.

Message Area

The Message Area is just above the Function Key Display at the bottom of the Context Manager screen. It prompts you with the status, options, and/or limitations associated with the context you are working on.

Message Line

The Message Line is located immediately below the Identification Line on the CM Editor screen. During your operation of CM Editor, messages display on this line telling you what you can or should do next.

Partition

A partition is a defined portion of workstation memory. Under Context Manager, BTOS resides in the low end of memory and Context Manager itself remains in the primary partition. As you begin each application, Context Manager places it in one of the remaining partitions.

Release Disk

The release disk includes the ability to invoke the Executive, WriteOne, and Logout applications. These commands come with the default configuration file. Additionally, Context Manager retains color and graphics qualities between contexts.

Remove (f6) Key

When you press this key at the CM Editor screen, the system removes the name in the Command Name field from the Command List Area.

Rename (f7) Key

When you press this key at the CM Editor screen, the system allows you to edit the command name displayed in the Command Name field.

Right and Left Arrow Keys

The **Right** and **Left Arrow** keys move the highlight on the Context Manager screen between the blocks labeled Applications You Can Start and Contexts You Can Return To. On the CM Editor screen, these keys move the cursor horizontally within a field.

Running Status Term

A status of Running means that the application is currently operating.

Show (f3) Key

When you press this key at the CM Editor screen, the system overwrites the CM Editor screen Function Key Display with the current values on the Context Manager screen Function Key Display.

Status Column

The Status column appears on the Context Manager screen to the left of the bullet. It indicates the standing of the corresponding context.

Status Terms

Status terms refer to the standing of a specific context. They are: running, waiting, done, swapped, and stopped. (See individual Glossary entries.)

Stopped Status Term

A status of Stopped means that the context is in the background, but not running. This context can run only in the foreground.

Swap (f8) Key

When you press this key at the CM Editor screen, the system prompts you to identify the swap file for this configuration file.

Swap File

A swap file is a hard disk file to which Context Manager can swap contexts and thus increase the number of available contexts. You have the option of creating a swap file. However, swapping increases the flexibility of Context Manager by allowing you to open up to ten applications at the same time.

Swapped Context

A swapped context refers to a context that Context Manager temporarily puts in your system's swap file, thus leaving room for you to open more applications when your system is full. The swapping capability and the extent of swapping is dependent upon the size of the configured swap file.

Swapped Status Term

A status of Swapped means that the system temporarily stores a context on disk to make room for another context. The swapped context is open and available to you, but is suspended until you recall it.

Tentatively Clean Context State

All contexts start in a tentatively clean state.

Tentatively Dirty Context State

Contexts in the tentatively clean state go to the tentatively dirty state if they do a ResetVideo. If the context is not on the screen and is marked tentatively dirty, the context is suspended.

Undo (f2) Key

When you press this key at the CM Editor screen, the system replaces the value in the highlighted field with the immediately previous value for that field. In numeric fields, if the previous value was blank, the value becomes 0.

Up and Down Arrow Keys

The **Up Arrow** and **Down Arrow** keys move the highlight vertically within each of the areas on the Context Manager and CM Editor screens.

Video Pointer Map

The video pointer map is a system structure for each memory partition. It is an array of pointers, one for each line of the screen, that always points to the location of the associated line of the application's screen.

Waiting Status Term

A status of Waiting means that the application is not operating; it is waiting for your input.

INDEX

Documentation Evaluation Form

Title: B 20 Systems Context Manager Reference Manual

Form No: 5016181

Date: March 1985

Burroughs Corporation is interested in receiving your comments and suggestions regarding this manual. Comments will be utilized in ensuing revisions to improve this manual.

Please check type of Suggestion:

☐ Addition ☐ Deletion ☐ Revision ☐ Error

Comments:

From:

Name _____

Title _____

Company _____

Address _____

_____

Phone Number _____ Date _____

Remove form and mail to:

Burroughs Corporation
Corporate Documentation – West
1300 John Reed Court
City of Industry, CA 91745
U.S.A.